openEuler开源创新实践课

LFS7.7-systemd on openEuler

实验指导手册

v1.2



华为技术有限公司

目录

[1 实验介绍 3](#_Toc94084292)

[1.1 简介 3](#_Toc94084293)

[1.2 内容描述 3](#_Toc94084294)

[1.3 读者知识背景 3](#_Toc94084295)

[1.4 实验环境说明 4](#_Toc94084296)

[1.4.1 个人电脑及操作系统 4](#_Toc94084297)

[1.4.2 虚拟机软件及其版本 4](#_Toc94084298)

[1.4.3 作为LFS宿主机的虚拟机规格 4](#_Toc94084299)

[1.4.4 openEuler操作系统镜像 5](#_Toc94084300)

[1.4.5 LFS版本 5](#_Toc94084301)

[1.5 参考资料 5](#_Toc94084302)

[2 宿主机系统准备 6](#_Toc94084303)

[2.1 虚拟机的创建与配置 6](#_Toc94084304)

[2.1.1 创建虚拟机 10](#_Toc94084305)

[2.1.2 对虚拟机的部分配置进行调整 15](#_Toc94084306)

[2.2 安装openEuler操作系统 30](#_Toc94084307)

[2.2.1 将openEuler20.09\_x86-64镜像装载到光驱 30](#_Toc94084308)

[2.2.2 安装操作系统 31](#_Toc94084309)

[2.3 准备实验环境 43](#_Toc94084310)

[2.3.1 宿主机软件环境准备 43](#_Toc94084311)

[2.3.2 宿主机硬件环境准备 44](#_Toc94084312)

[3 从社区获取资料 51](#_Toc94084313)

[3.1 注册gitee账户 51](#_Toc94084314)

[3.2 从git仓库获取资料 52](#_Toc94084315)

[3.3 将资料复制到本地 52](#_Toc94084316)

[4 LFS系统的构建 54](#_Toc94084317)

[4.1 准备构建LFS系统所需的包和补丁 54](#_Toc94084318)

[4.2 创建并以lfs用户编译临时工具链 54](#_Toc94084319)

[4.2.1 /mnt/lfs及lfs用户 54](#_Toc94084320)

[4.2.2 $LFS/sources 和$LFS/tools文件夹 55](#_Toc94084321)

[4.2.3 通用编译指南 55](#_Toc94084322)

[4.2.4 gcc-4.9.2编译问题 55](#_Toc94084323)

[4.2.5 其他需要注意的地方 55](#_Toc94084324)

[4.3 在chrooted环境下构建真正的LFS目标系统 56](#_Toc94084325)

[4.3.1 mount & chroot 56](#_Toc94084326)

[4.3.2 通用编译指南 56](#_Toc94084327)

[4.3.3 第二个版本的chroot 57](#_Toc94084328)

[4.3.4 其他问题 57](#_Toc94084329)

[4.4 配置LFS目标系统 57](#_Toc94084330)

[4.5 编译并安装内核 58](#_Toc94084331)

[4.6 设置GRUB使新系统可引导 59](#_Toc94084332)

[4.6.1 LFS系统GRUB设置 59](#_Toc94084333)

[4.6.2 宿主机GRUB设置 59](#_Toc94084334)

[4.7 进入新的系统 60](#_Toc94084335)

[5 作业的提交 62](#_Toc94084336)

[6 缩略语表 63](#_Toc94084337)

# 实验介绍

## 简介

本手册为openEuler开源创新实践课实验教程，适用于已有一定计算机原理、操作系统、编译原理基础，并希望深入了解操作系统及相关工具的构建，从源码开始搭建个性化操作系统的读者。

鉴于创新实践课本身的特点，本文不会给出一幅按图索骥的地图或一系列按部就班的步骤来指导你构建一个新系统，而是将“如何构建一个自己的Linux操作系统”的津要和盘托出。即使是这样，依托我们专门撰写的“构建脚本”，足以使您能够顺利地完成构建LFS系统的基础工作。

## 内容描述

本实验指导书共包含4个部分，从实验简介到宿主机环境准备，再到构建LFS系统，最后是供学生完全自己发挥的拓展实验。

## 读者知识背景

本课程为操作系统进阶实践课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

* 有电脑操作基础
* 熟悉Linux操作系统的基本架构和实现原理
* 掌握Linux操作系统的基本操作
* 了解Linux常用工具的功能和使用

## 实验环境说明

### 个人电脑及操作系统

每套实验环境包括个人PC一台，每套实验环境适用于1名学员上机操作。为了满足实验需要，建议每套实验环境采用以下配置：

实验设备

|  |  |  |
| --- | --- | --- |
| **设备名称** | **设备型号** | **操作系统** |
| 个人电脑 | 处理器：Intel i7 CPU  逻辑处理器数：≥6  内存：16G  系统类型：x64-based处理器 | Windows 10 / macOS |

注：此次实验需要的计算资源全部来自学员个人电脑，硬件上应尽量接近或高于上表配置以增进构建速度。

### 虚拟机软件及其版本

虚拟机软件

|  |  |  |
| --- | --- | --- |
| **软件名称及本实验推荐版本** | **官方网址** | **备注** |
| VMware Workstation 15.5 Pro | <https://www.vmware.com/> | 需付费使用 |
| Oracle VM VirtualBox 6.1 | <http://www.virtualbox.org> | 根据其相关协议使用 |

注：本实验所用虚拟机软件是Oracle VM VirtualBox 6.1。

### 作为LFS宿主机的虚拟机规格

虚拟机规格

|  |  |
| --- | --- |
| **参数** | **推荐值** |
| 逻辑处理器数 | 4 |
| 内存 | 8GB |
| 磁盘大小 | 30GB |

### openEuler操作系统镜像

推荐使用openEuler 20.03 LFS SP1或openEuler 20.09，本实验我们选择了后者。

openEuler操作系统镜像

|  |  |  |
| --- | --- | --- |
| **软件名称** | **CPU架构** | **下载地址** |
| openEuler 20.09 | x86\_64 | <https://repo.openeuler.org/openEuler-20.09/ISO/x86_64/openEuler-20.09-x86_64-dvd.iso> |

### LFS版本

本实验以LFS-BOOK-7.7-systemd为例进行讲解。为什么呢？因为该版本所用的GCC版本不是太高（gcc-4.9.2），这对于想从源码研究C++的人是一大福音，另外对systemd感兴趣的读者也可以从源码研究systemd。

LFS7.7-systemd

|  |  |
| --- | --- |
| **项目名称** | **下载链接** |
| Packages | <http://ftp.osuosl.org/pub/lfs/lfs-packages/lfs-packages-7.7-systemd.tar> 或者 <https://zhuanyejianshe.obs.cn-north-4.myhuaweicloud.com/chuangxinshijianke/lfs-packages-7.7-systemd.tar> |
| LFS-BOOK | <https://www.linuxfromscratch.org/lfs/downloads/7.7-systemd/> |
| Chinese LFS-BOOK | <https://linux.cn/lfs/LFS-BOOK-7.7-systemd/> |

## 参考资料

1. LFS官网：https://www.linuxfromscratch.org/lfs/

2. openEuler社区：https://openeuler.org/

3. build-lfs7.7systemd.sh构建脚本

# 宿主机系统准备

所谓**宿主机系统**系指创建LFS目标系统的宿主，具体而言，即虚拟机上的openEuler操作系统。本章介绍如何使用VirtualBox软件创建虚拟机、安装openEuler操作系统以及安装后的实验环境准备工作。

## 虚拟机的创建与配置

本节假定您已经安装好Oracle VM VirtualBox 6.1软件并以此为例讲解如何创建满足创建LFS系统的虚拟机。安装该虚拟机软件时一路默认安装即可，可以参考以下步骤：

Text

Description automatically generated with medium confidence

点击“下一步”。

Graphical user interface, text, application

Description automatically generated

保持默认设置，点击“下一步”。

Graphical user interface, text, application, email

Description automatically generated

可以保持默认设置，点击“下一步”。

Text

Description automatically generated

选择“Yes（同意）”。

Graphical user interface, text, application, email

Description automatically generated

这时，点击“安装”。

Graphical user interface, text, application

Description automatically generated

等待安装完毕。

A picture containing graphical user interface

Description automatically generated

安装完毕后点击“完成”按钮。

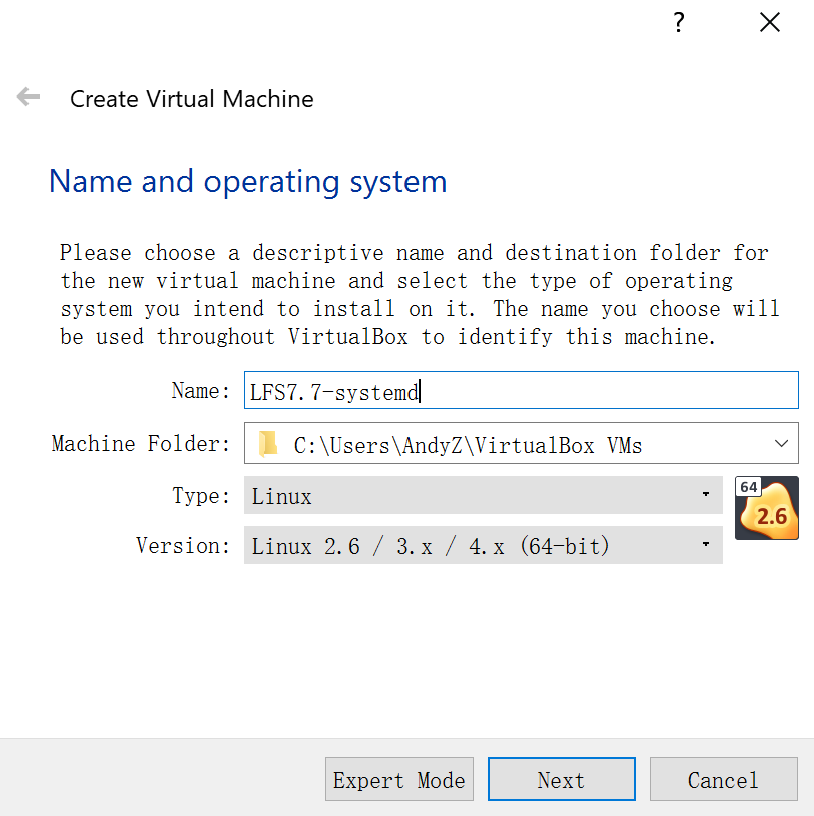
Graphical user interface, text, application

Description automatically generated

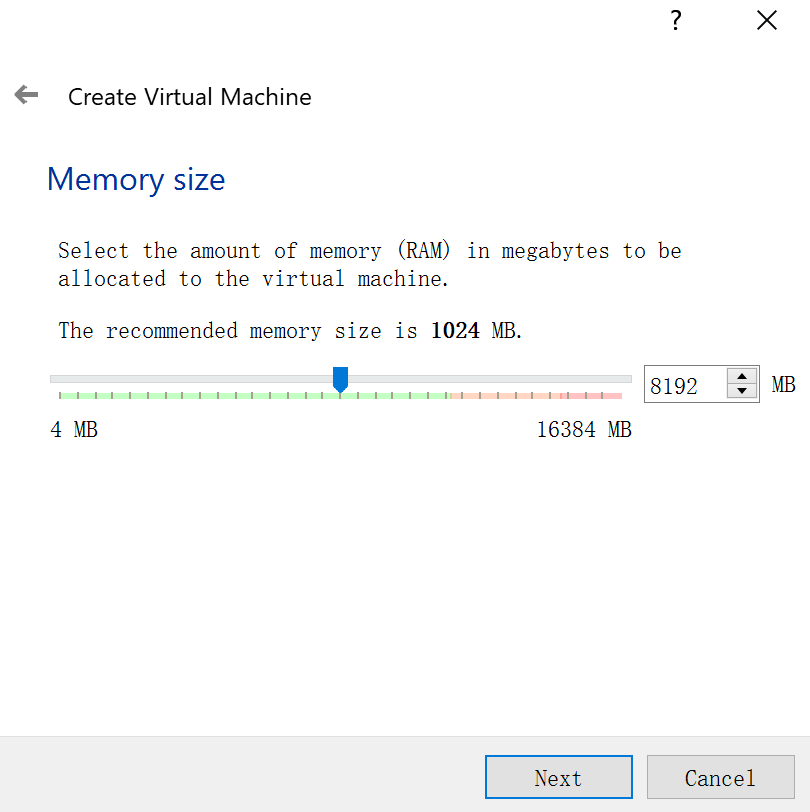
这是虚拟机软件启动之后的界面。

### 创建虚拟机

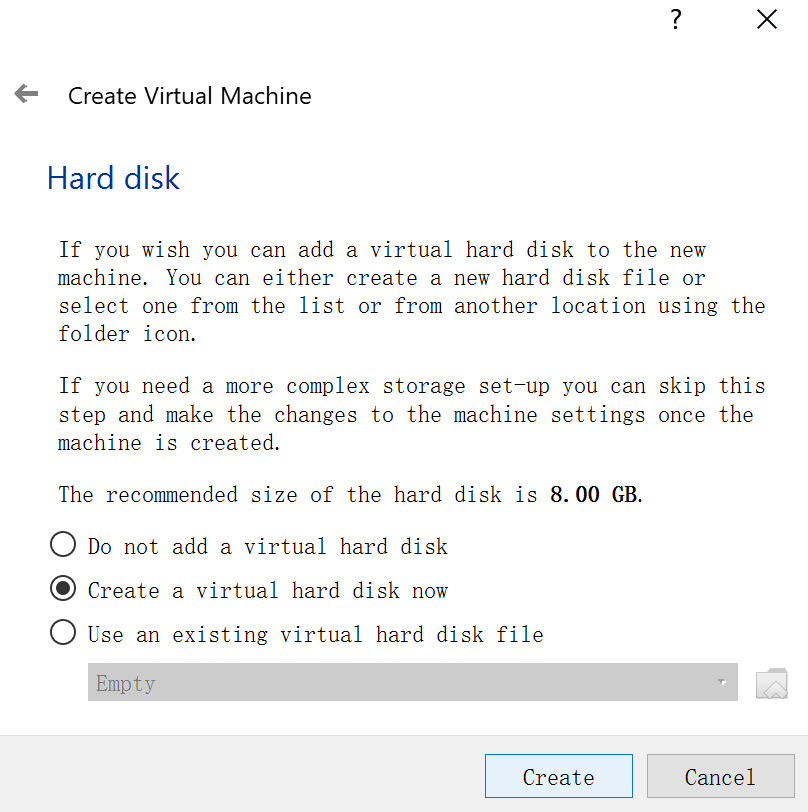
创建虚拟机时，请选择Linux类型和64-bit版本：



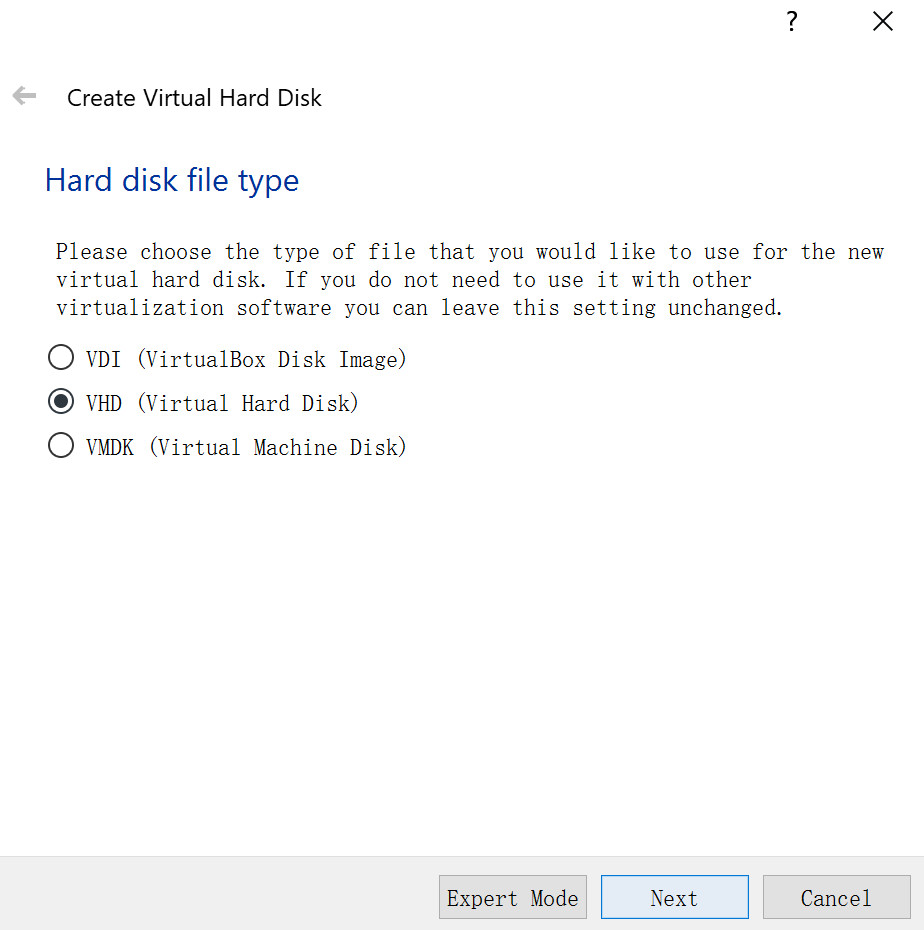
为了在构建LFS系统时有较快的速度，内存设置为8GB（或以上）：



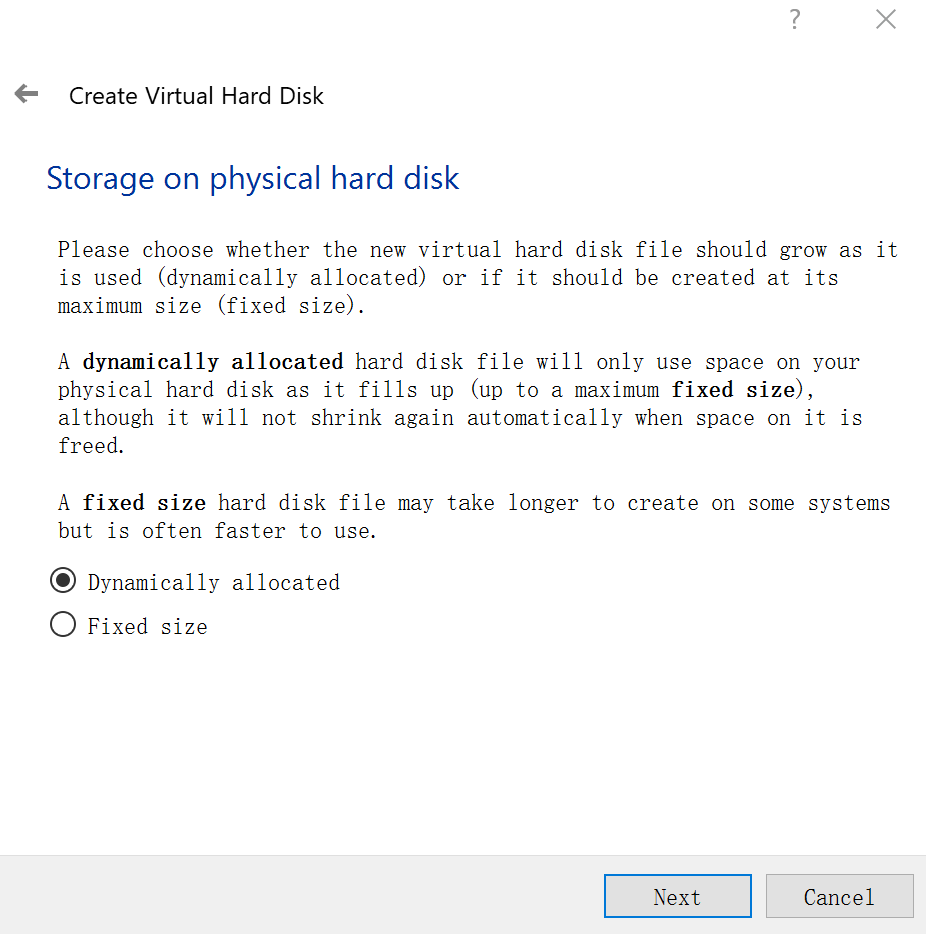
现在就创建硬盘：



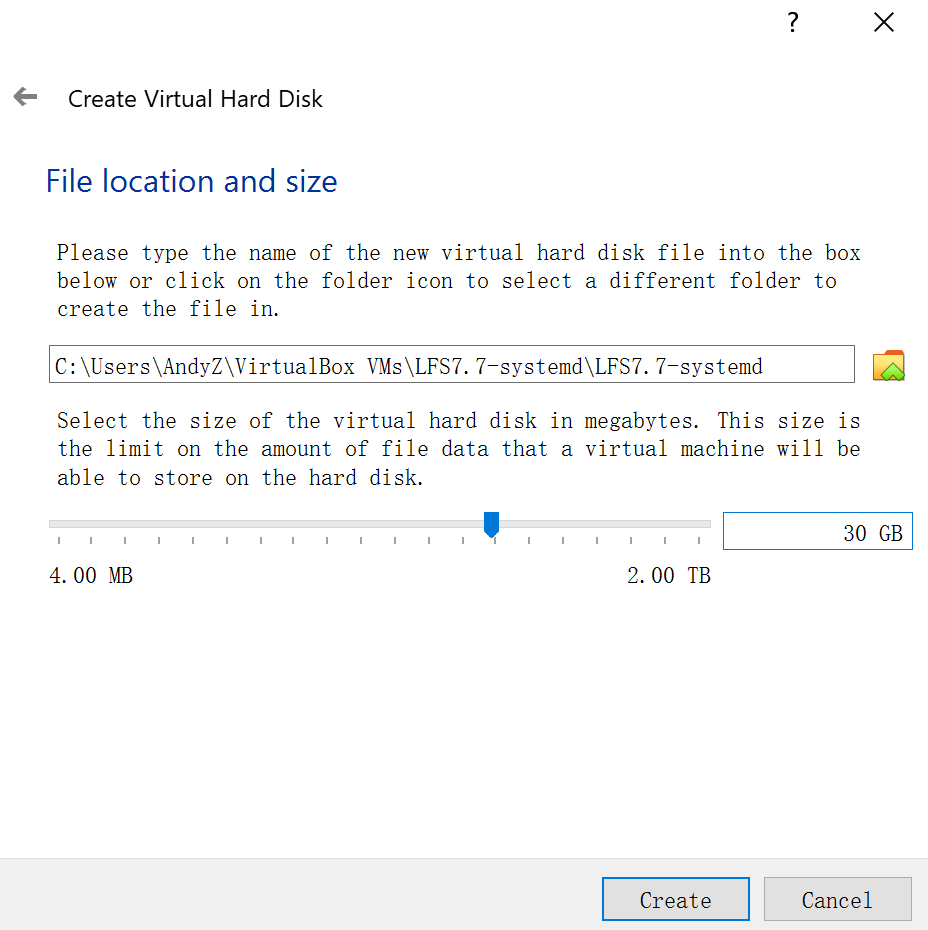
这里硬盘类型为VHD：



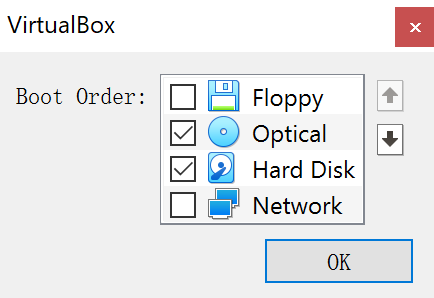
磁盘容量“动态增长”：



为保证足够的空间，为宿主机配置30GB大小的容量：

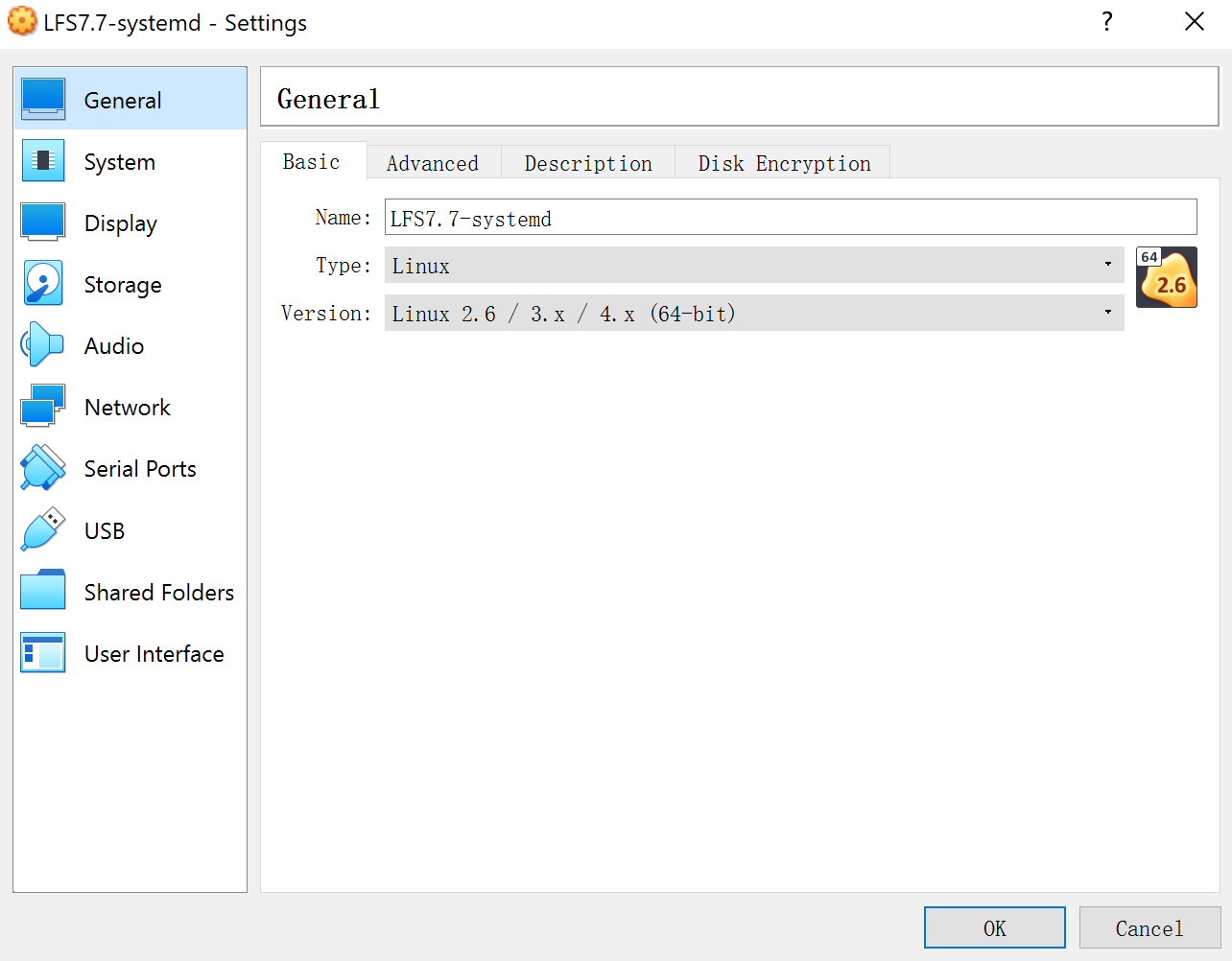


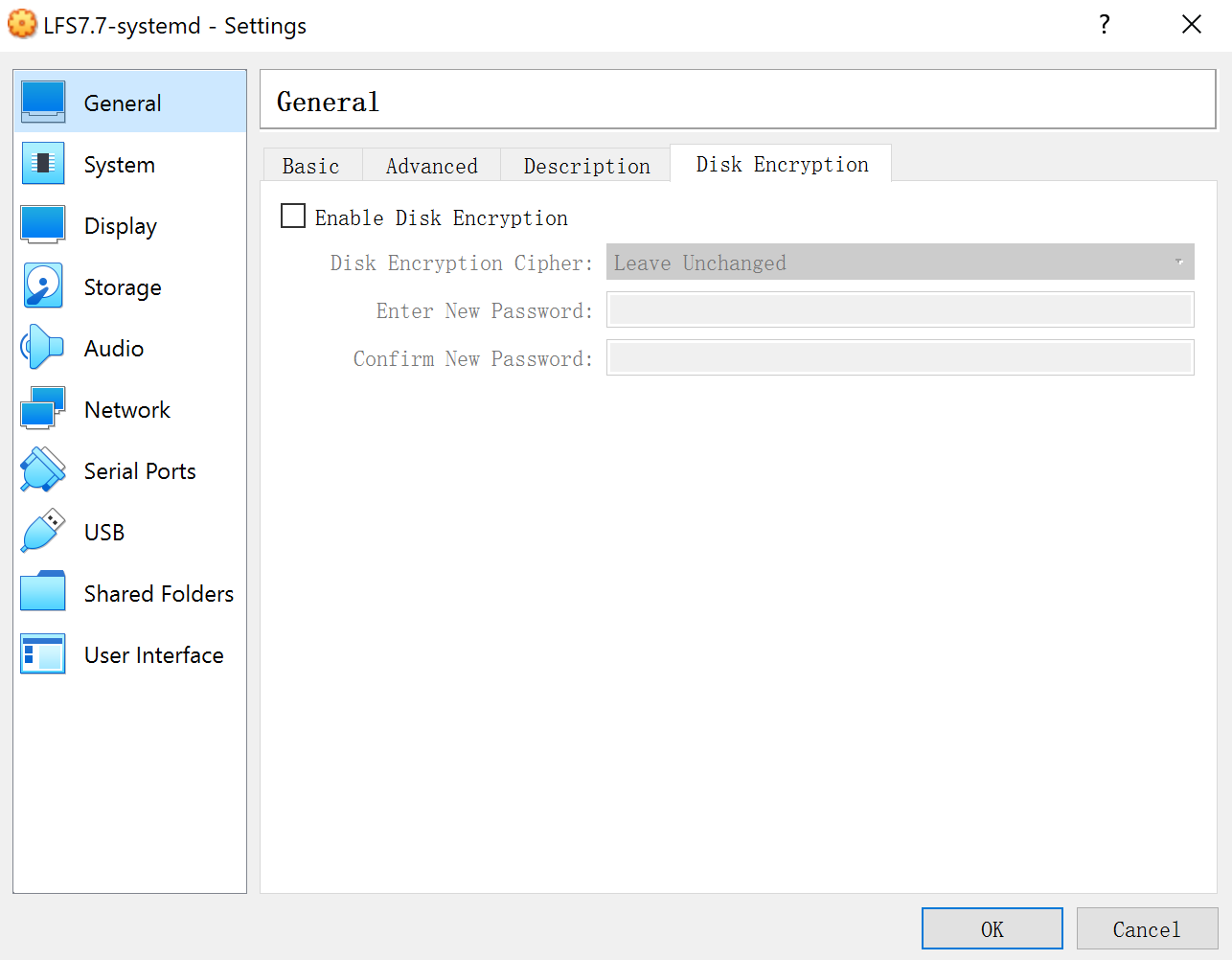
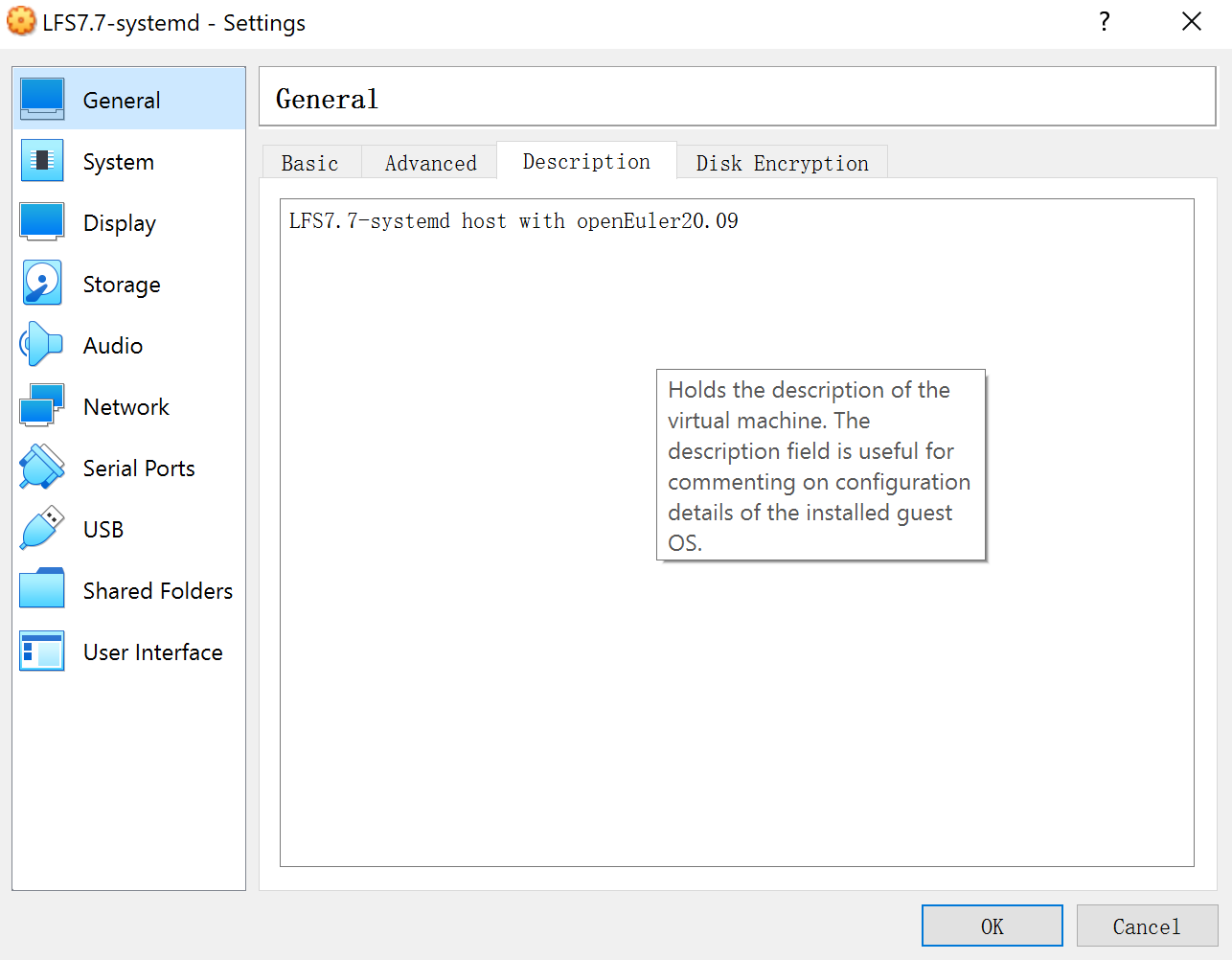
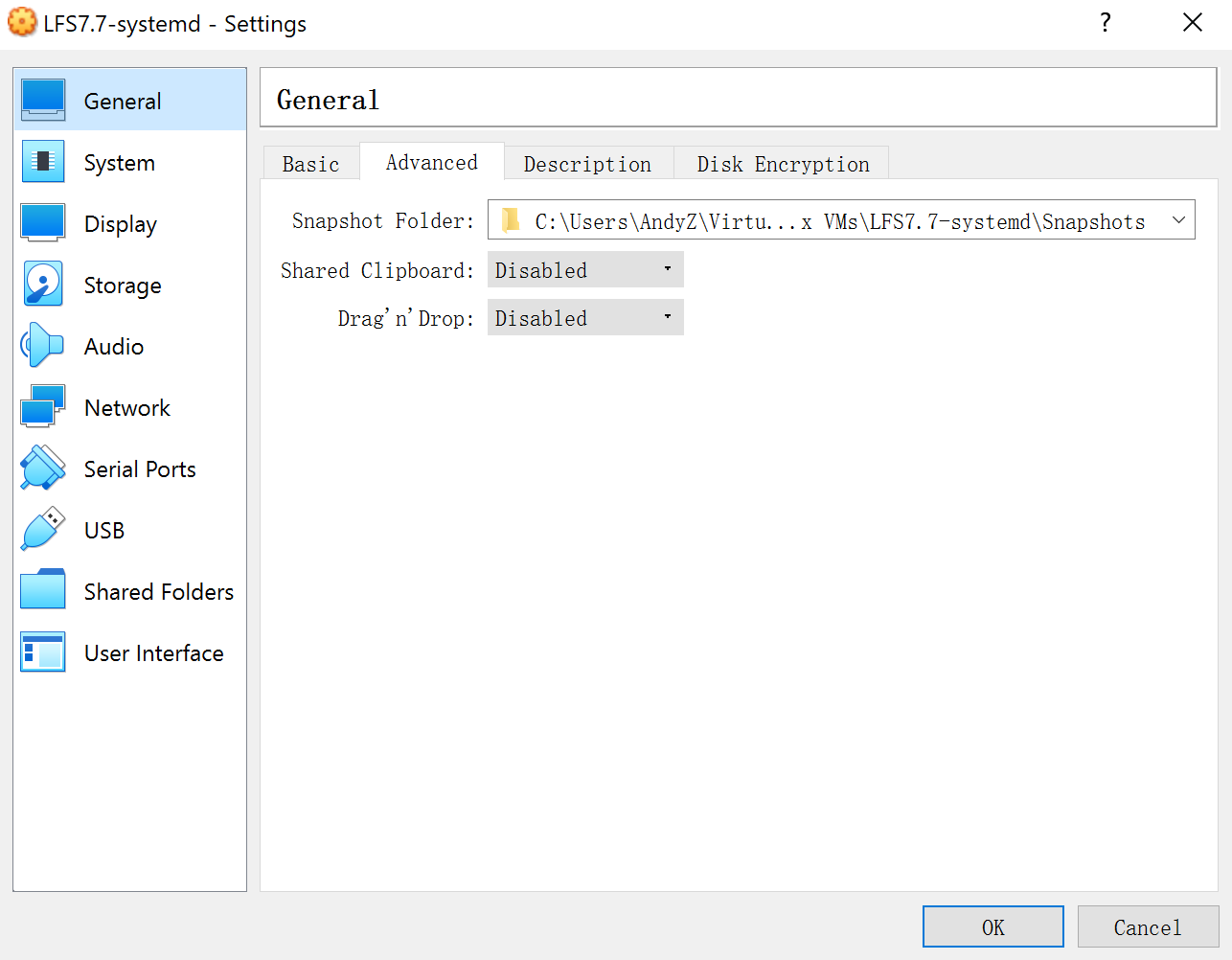
仅保留从光驱和硬盘启动：



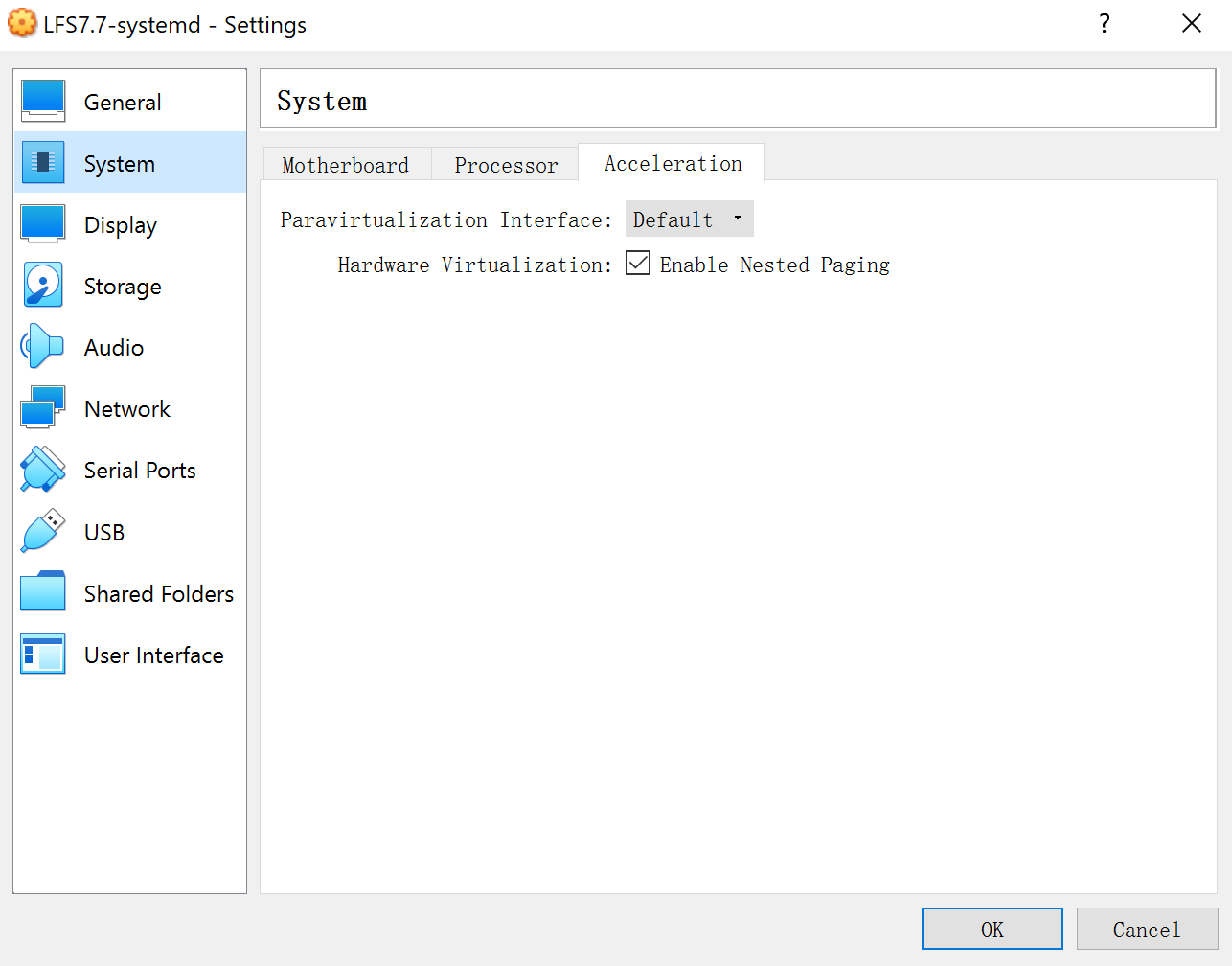
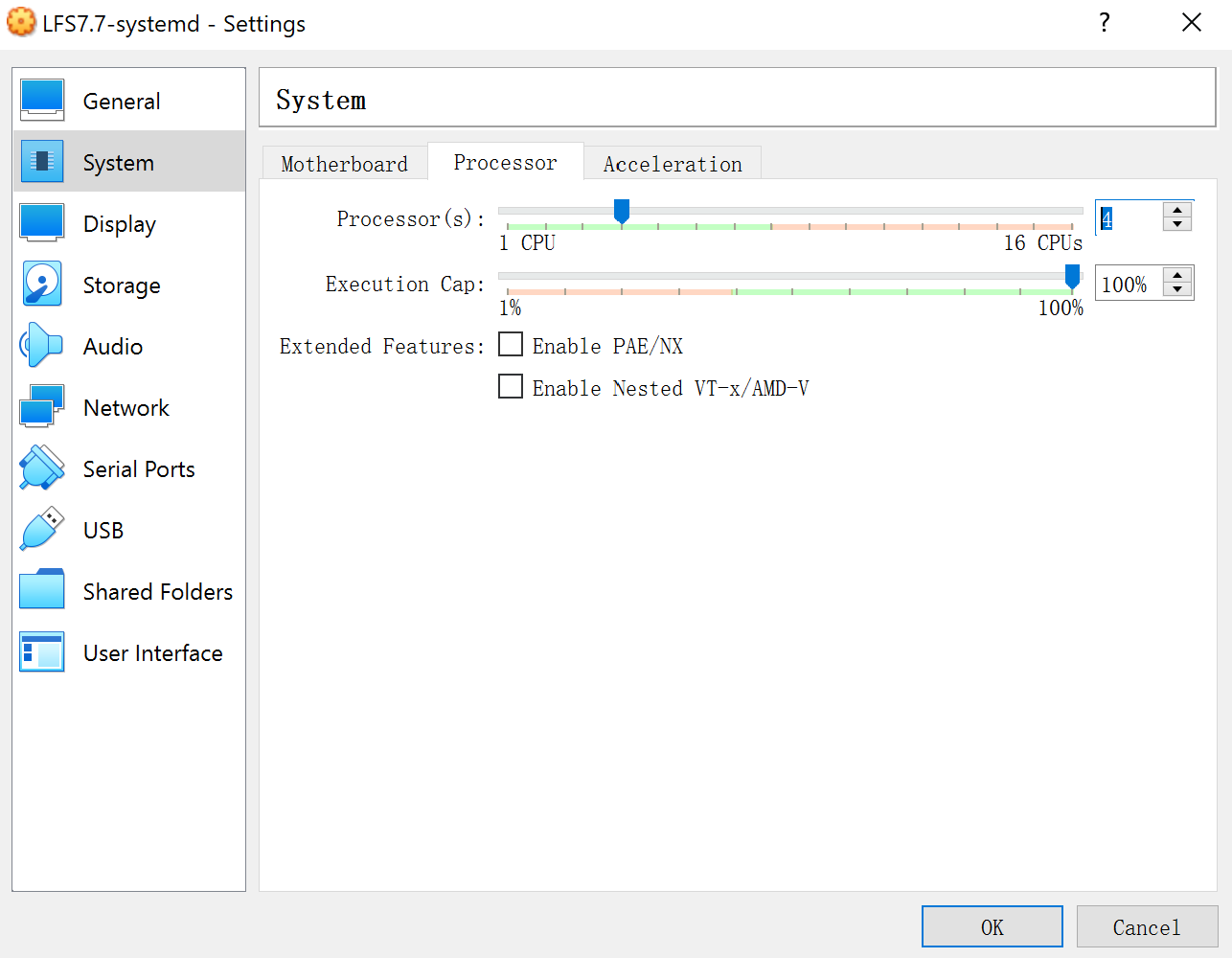
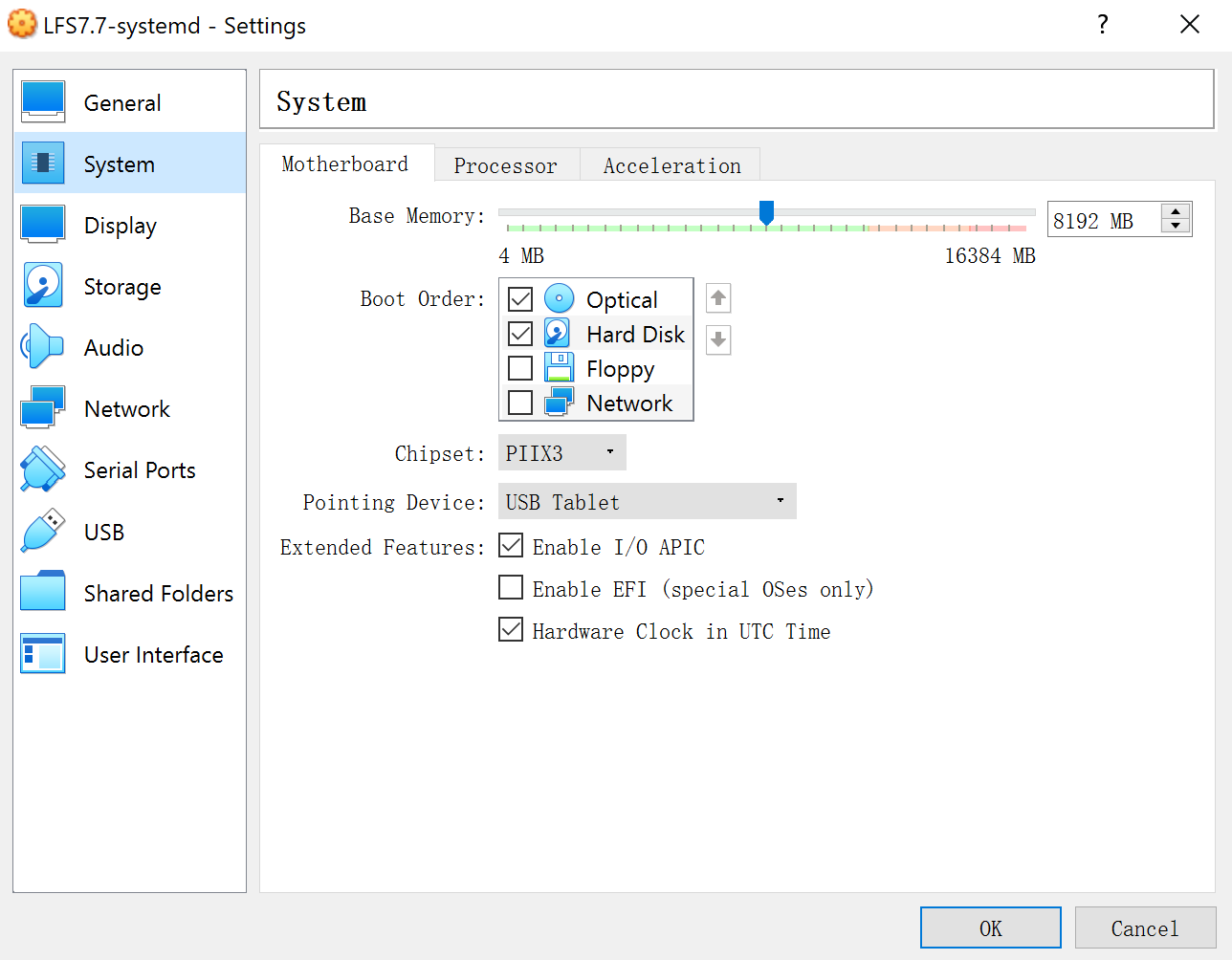
### 对虚拟机的部分配置进行调整

基本信息概览：

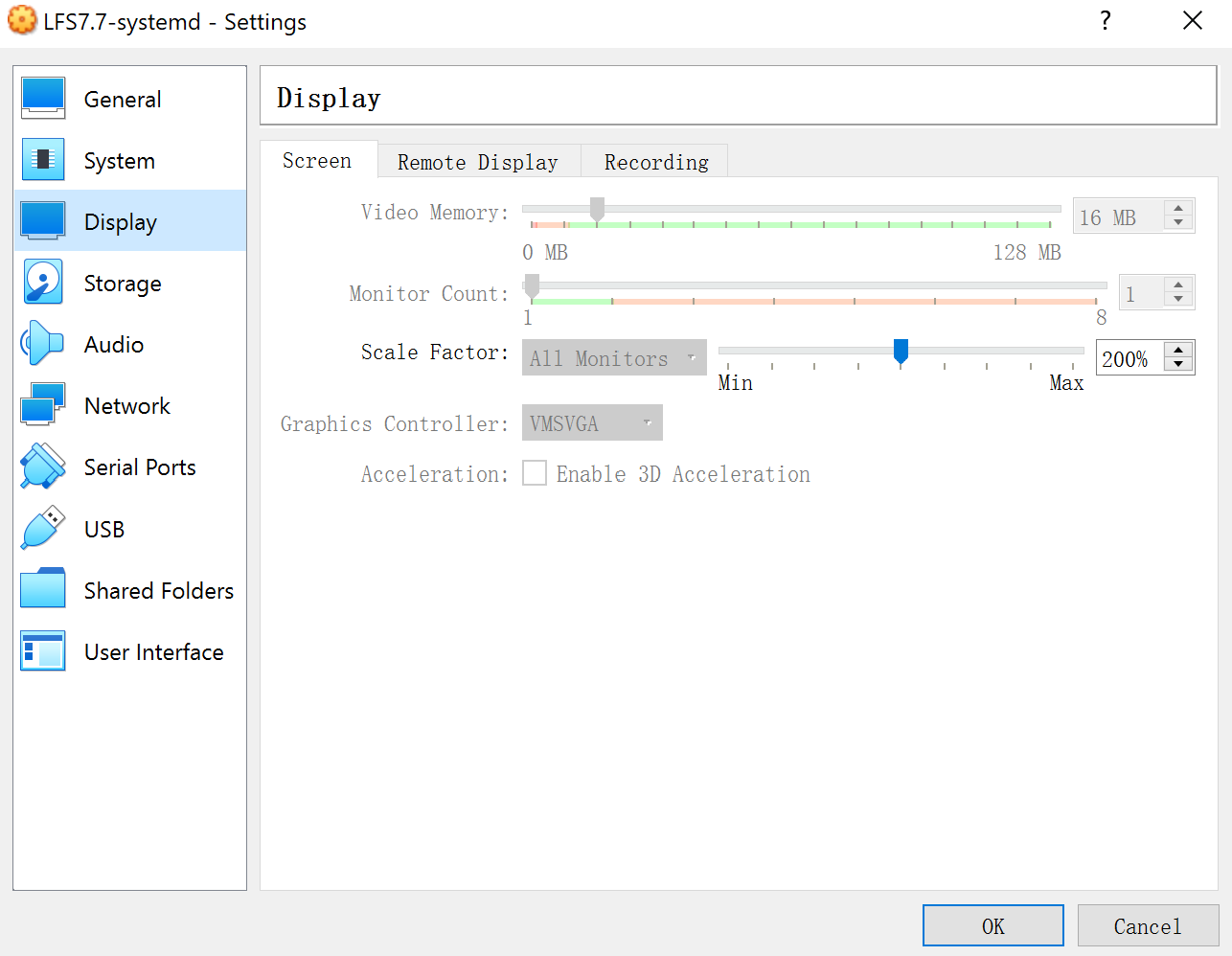




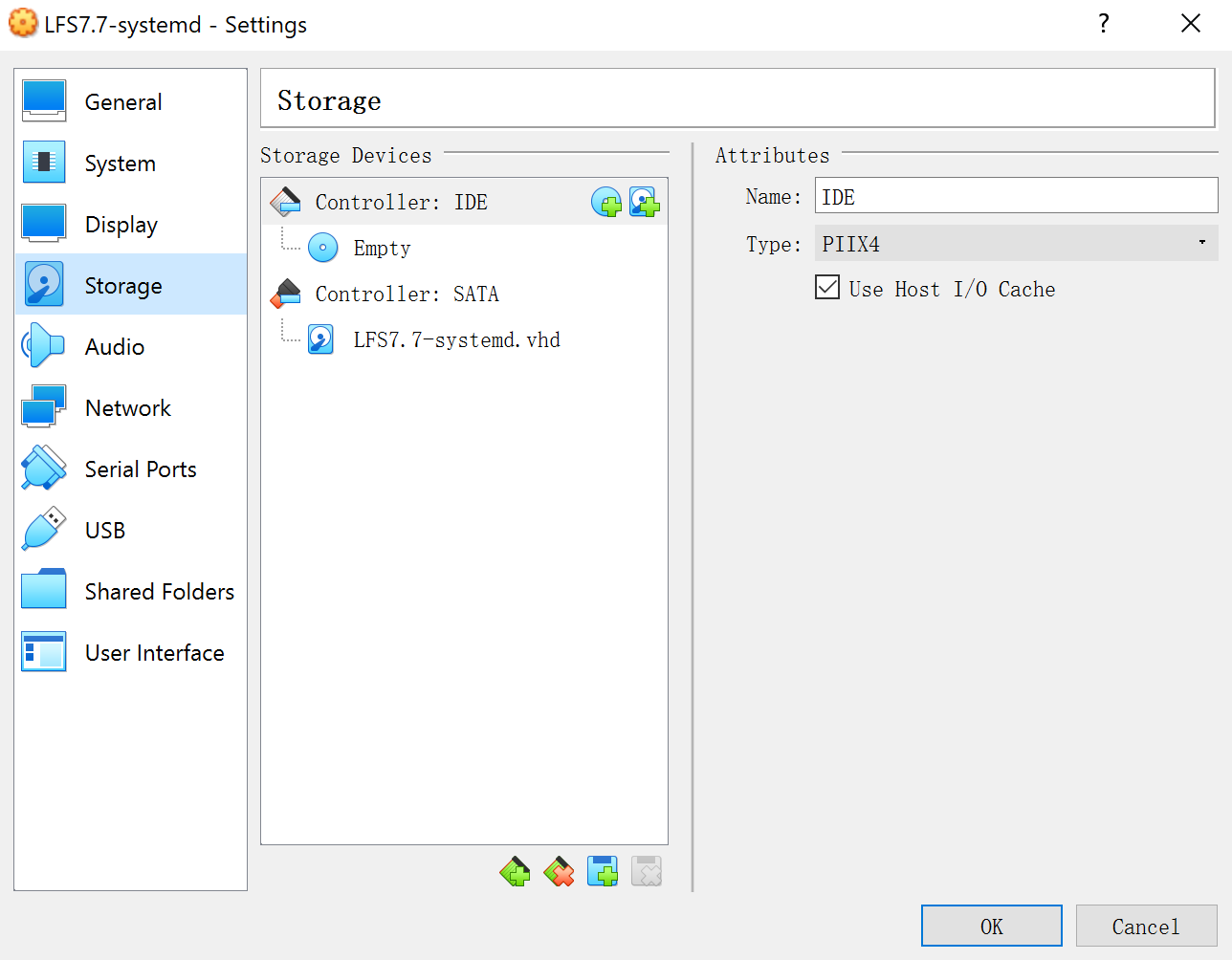
系统信息概览（注意将CPU至少设置成4核以提高构建系统的速度）：



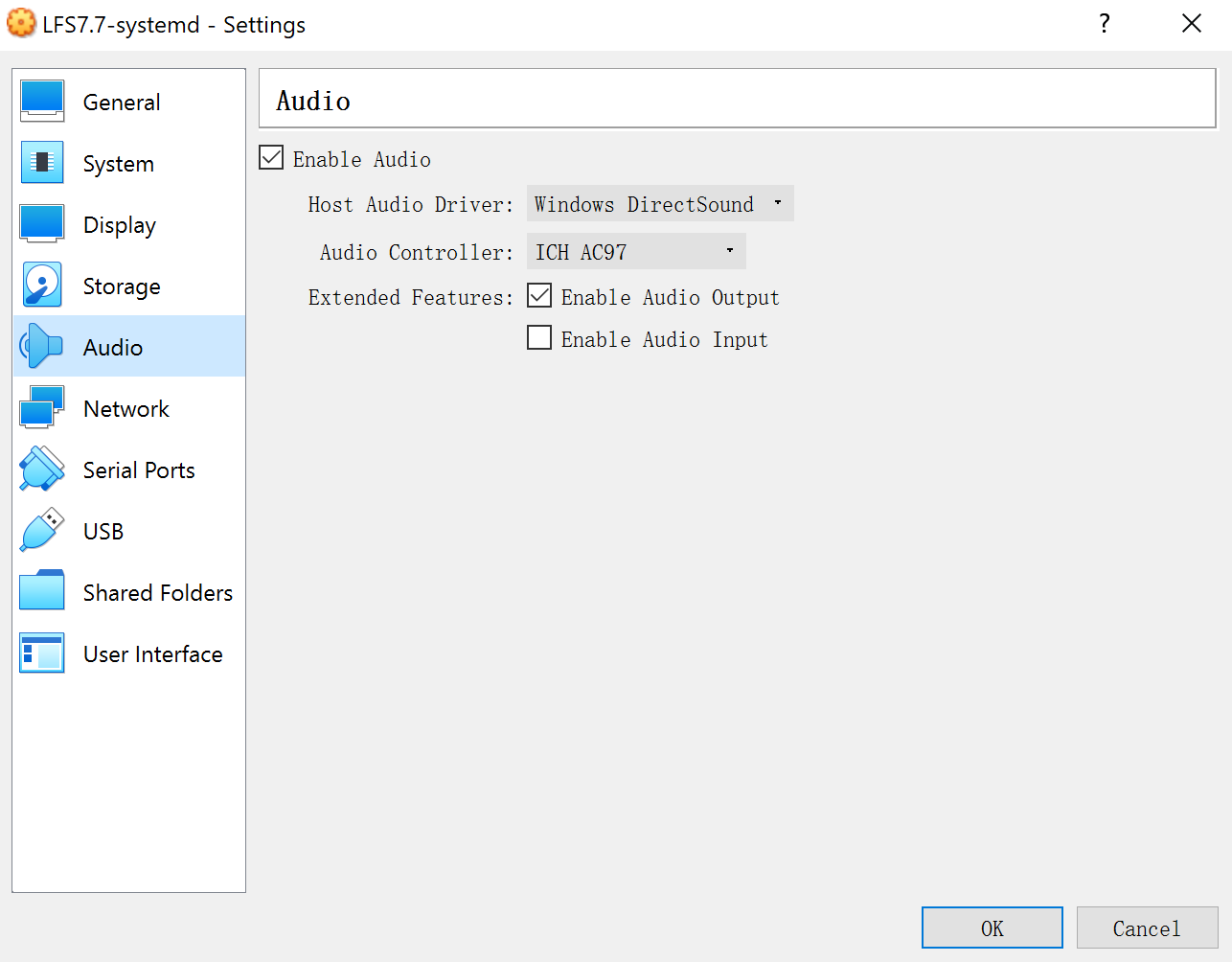
将显示器设置成200%的延展以获得较好的显示大小（请根据自己显示器实际分辨率设置此参数）：



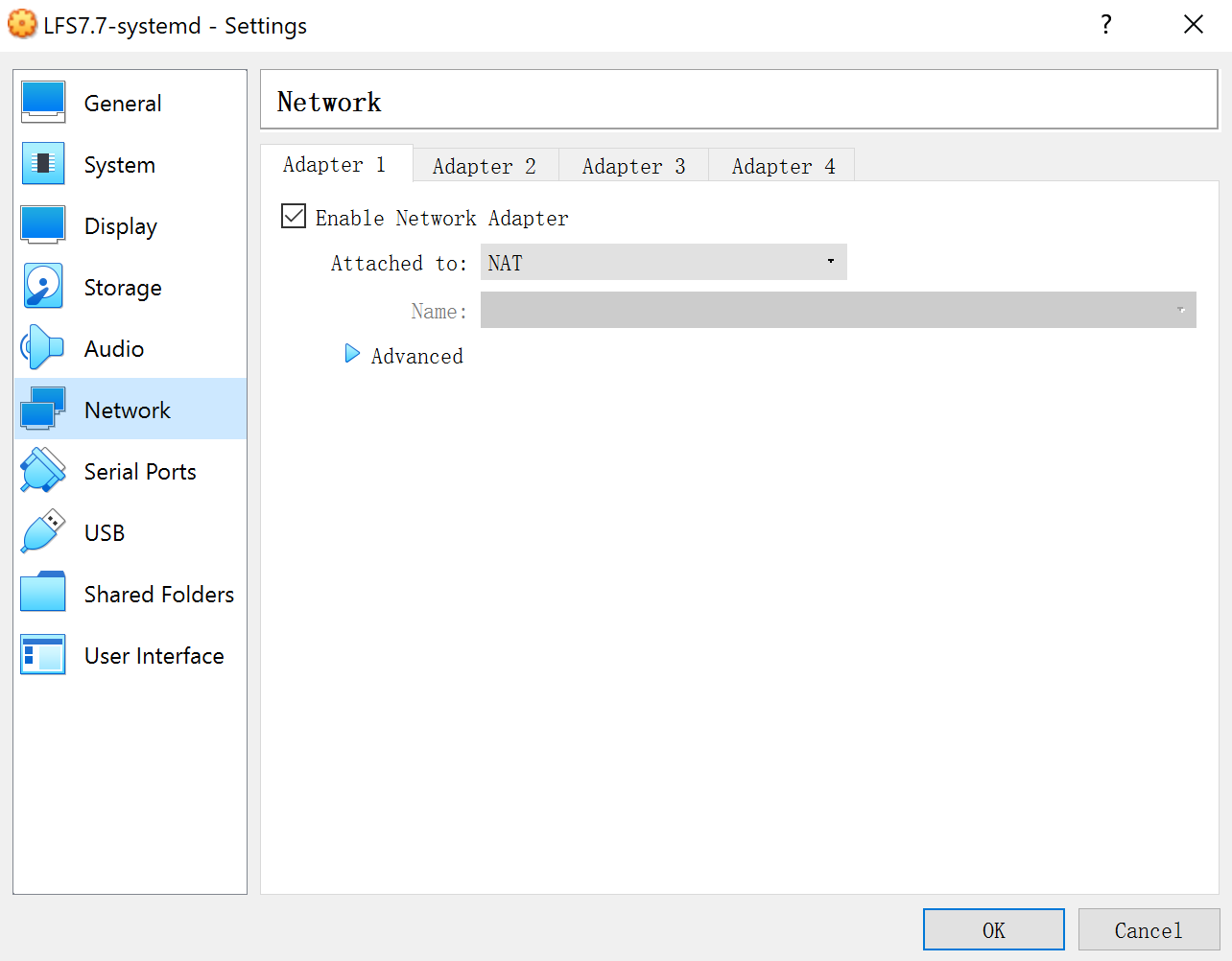
硬盘和光驱：



声音设备保持默认设置：



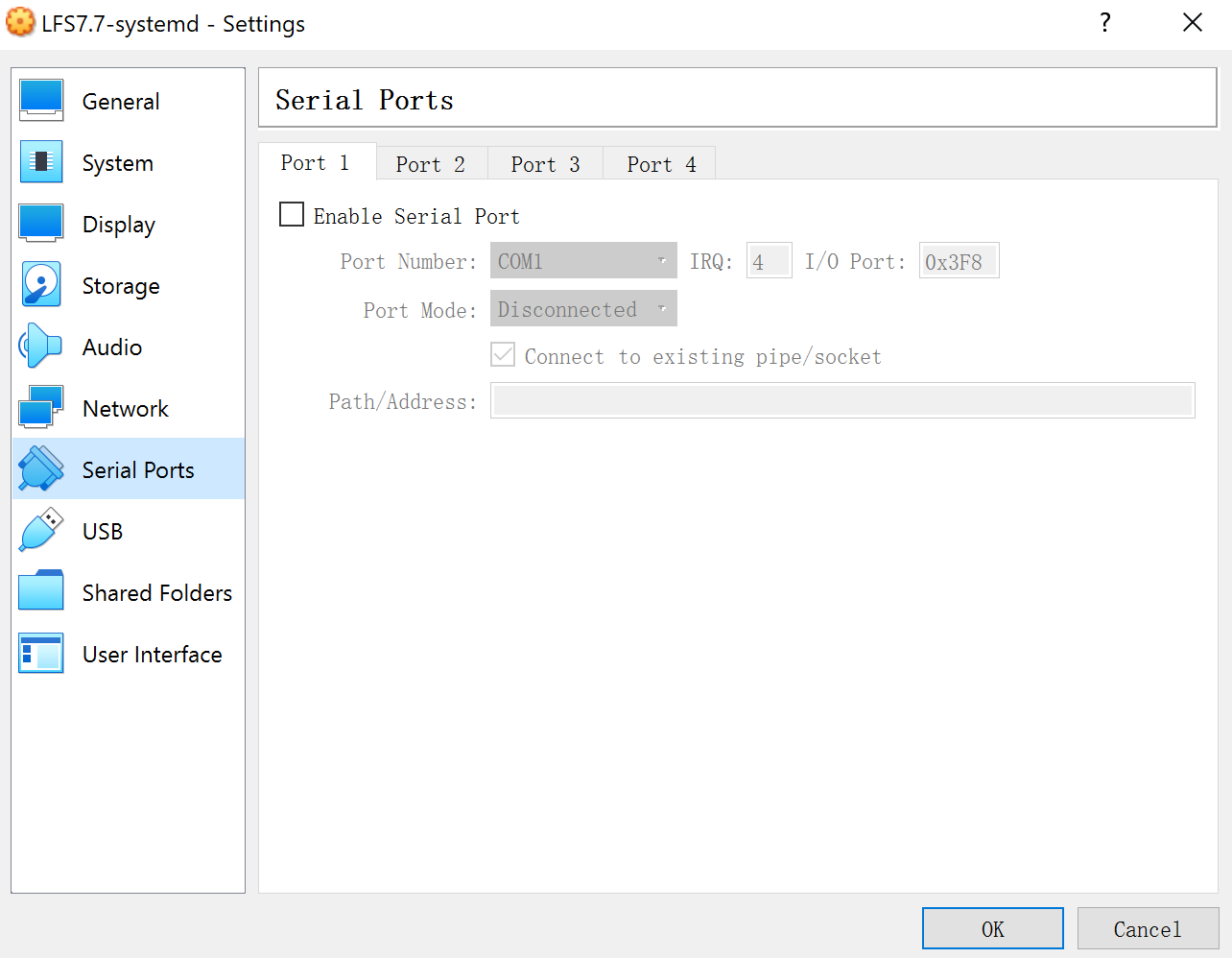
设置两张网卡，一张时默认的NAT模式（以便从虚拟机中访问外网）：



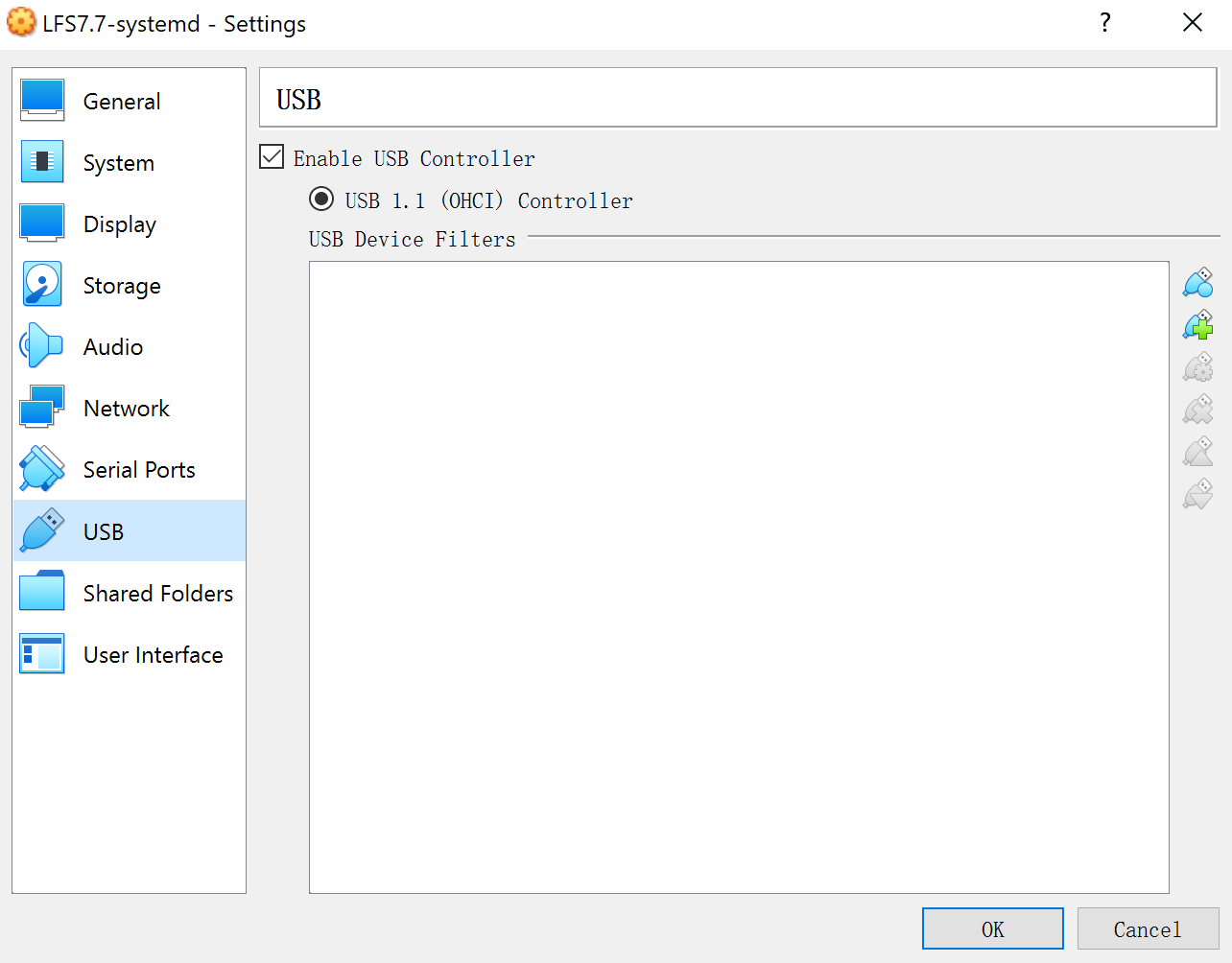
一张是Host-only模式（以便从该虚拟机的宿主机访问它）：



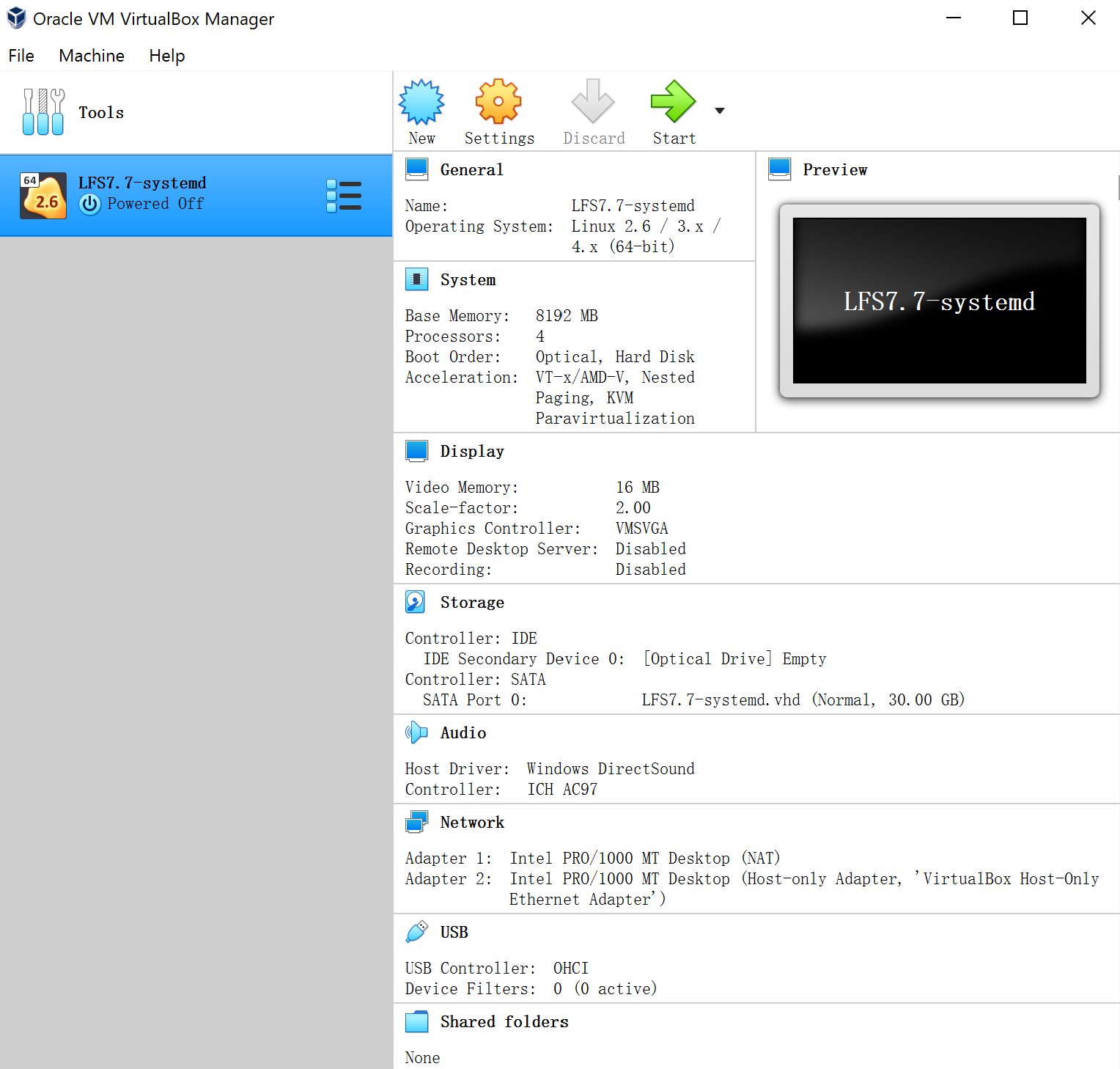
串口设置保持默认：



USB设置保持默认：



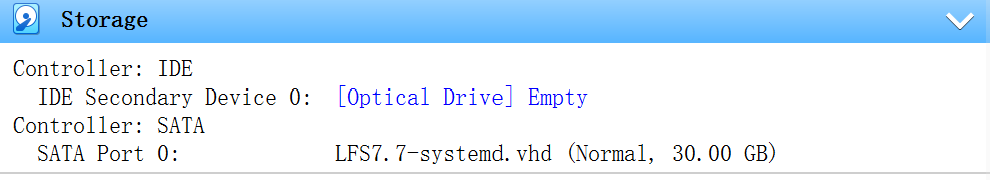
这是此时的整个虚拟机概况：



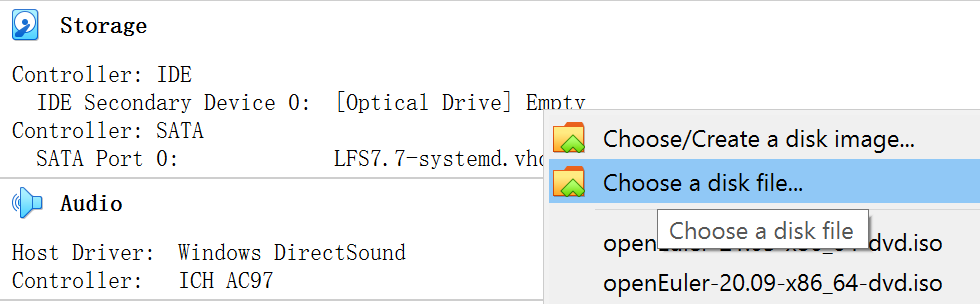
## 安装openEuler操作系统

### 将openEuler20.09\_x86-64镜像装载到光驱

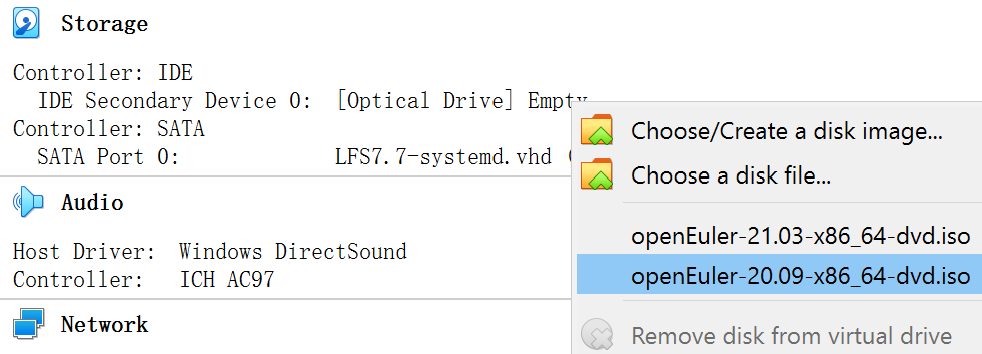
温馨提示：这里需要事先将openEuler20.09\_x86-64镜像下载到本地。



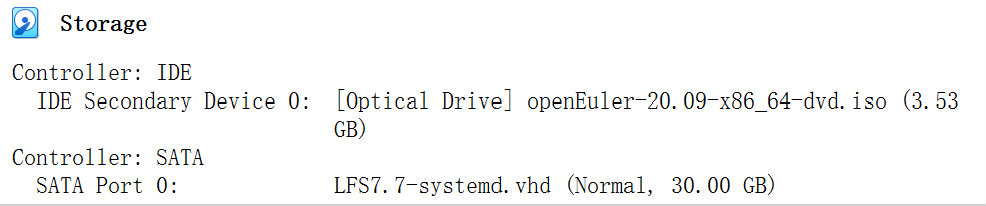
点击如上图所示“Optical Drive”（光驱）文字按钮。



点击“选择一个磁盘文件”然后选择先前自己下载到本地磁盘的openEuler安装镜像。



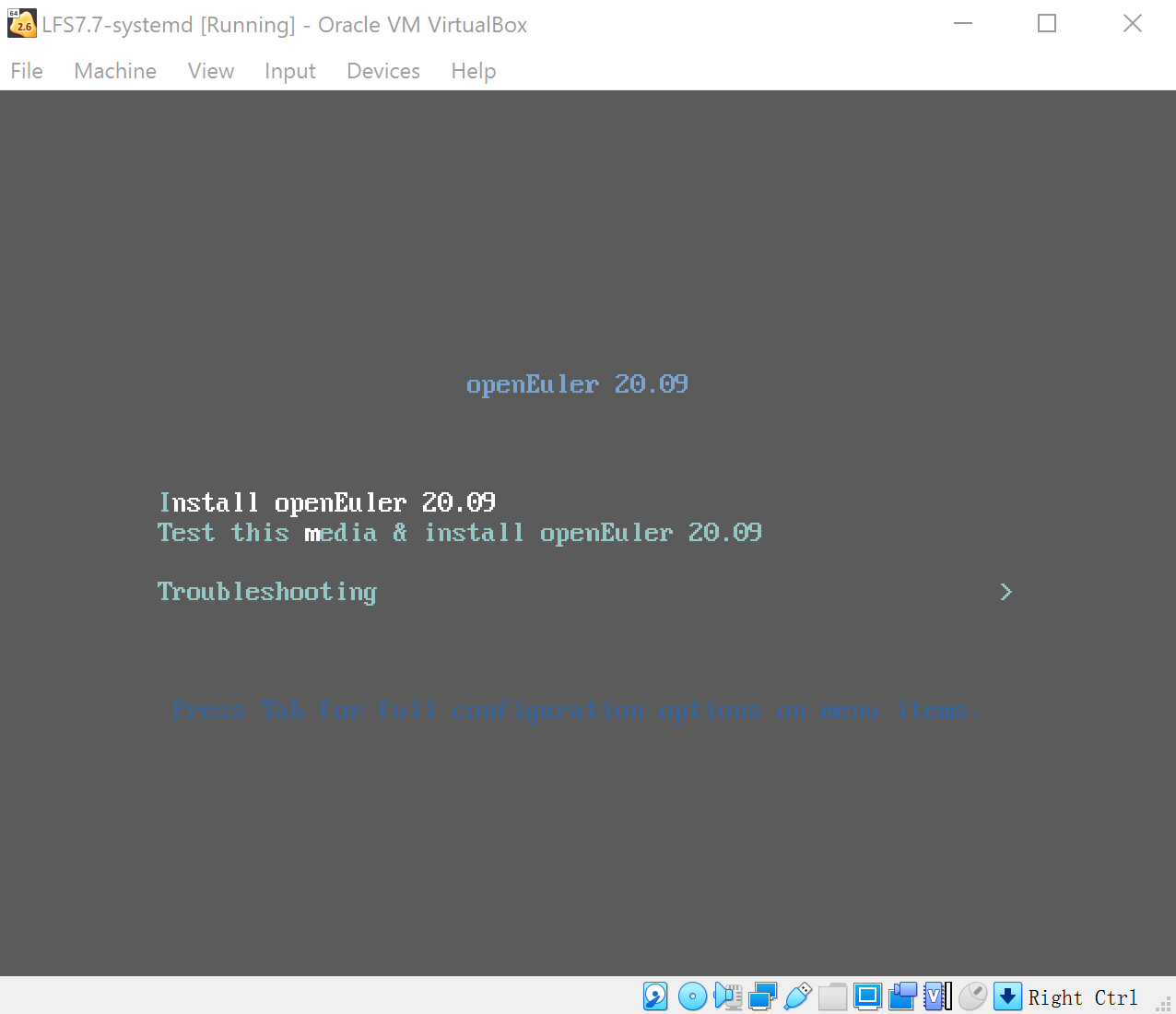
如果您已经执行过这个操作，则在菜单项中将有该镜像的名称，选中该名称的菜单项即可。



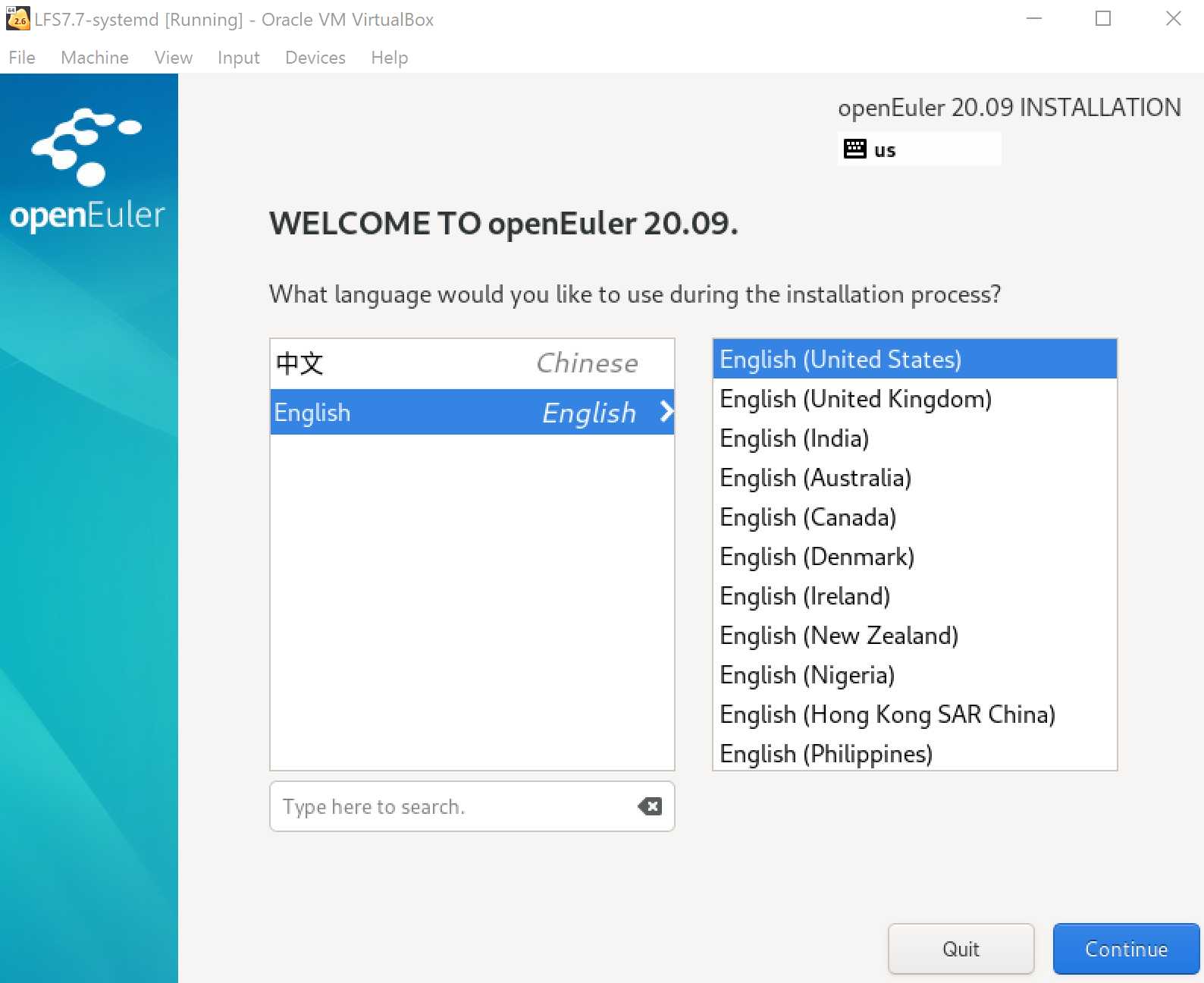
操作完毕后光驱中会显示该镜像的名称。

### 安装操作系统

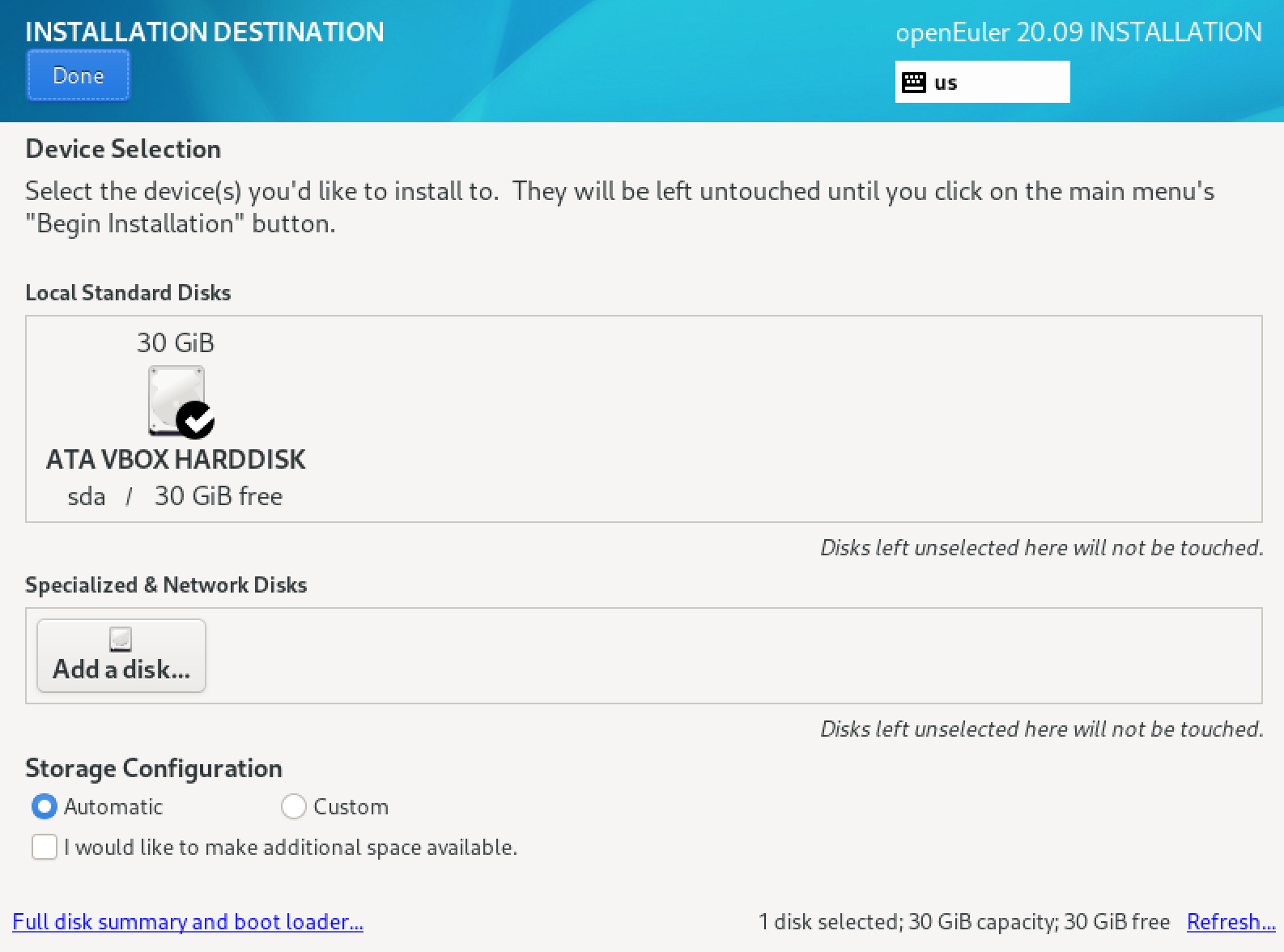
启动虚拟机，移动键盘的上下键选择“Install openEuler 20.09”菜单项并按下回车键进入安装界面：



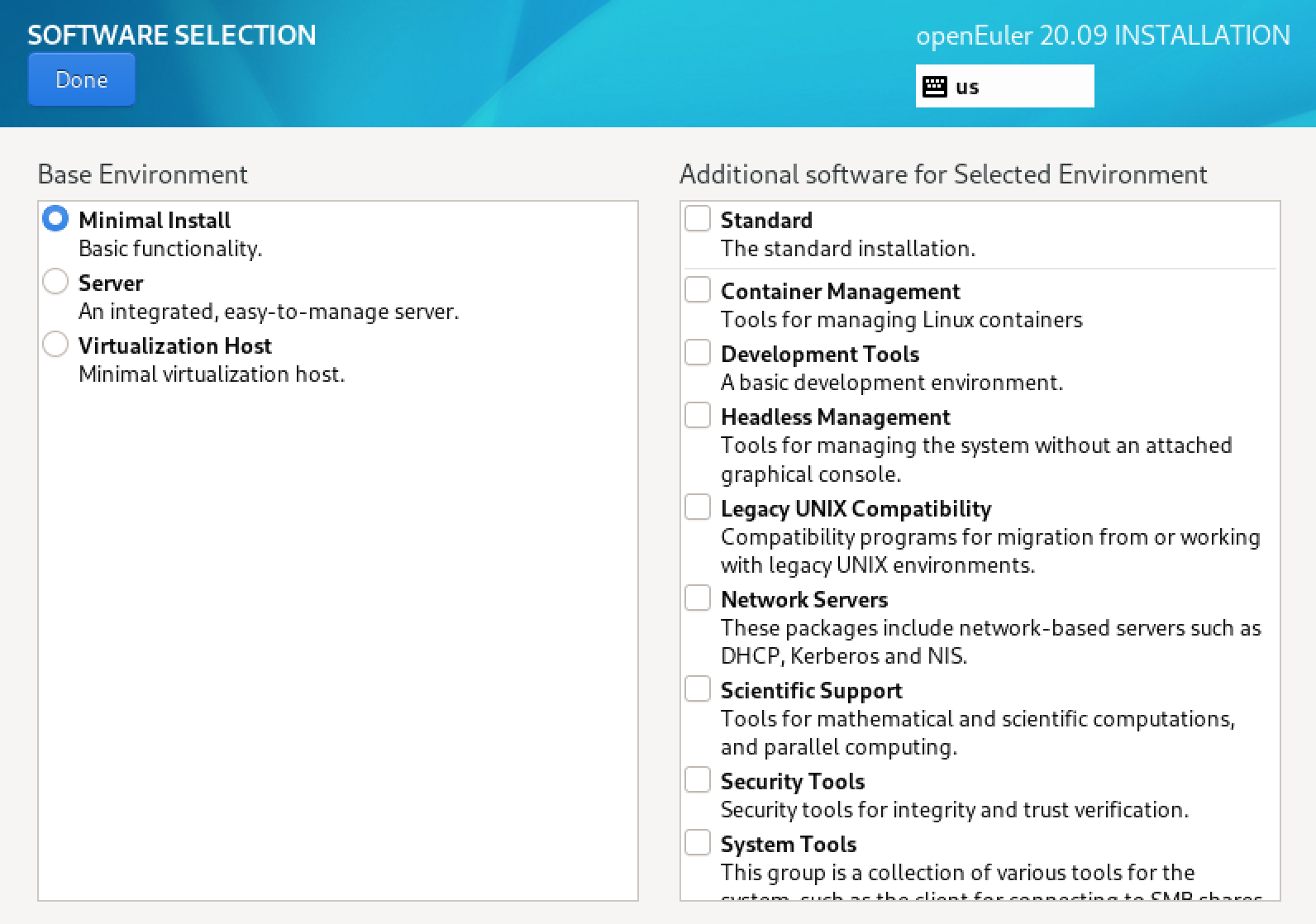
选择英文版安装：



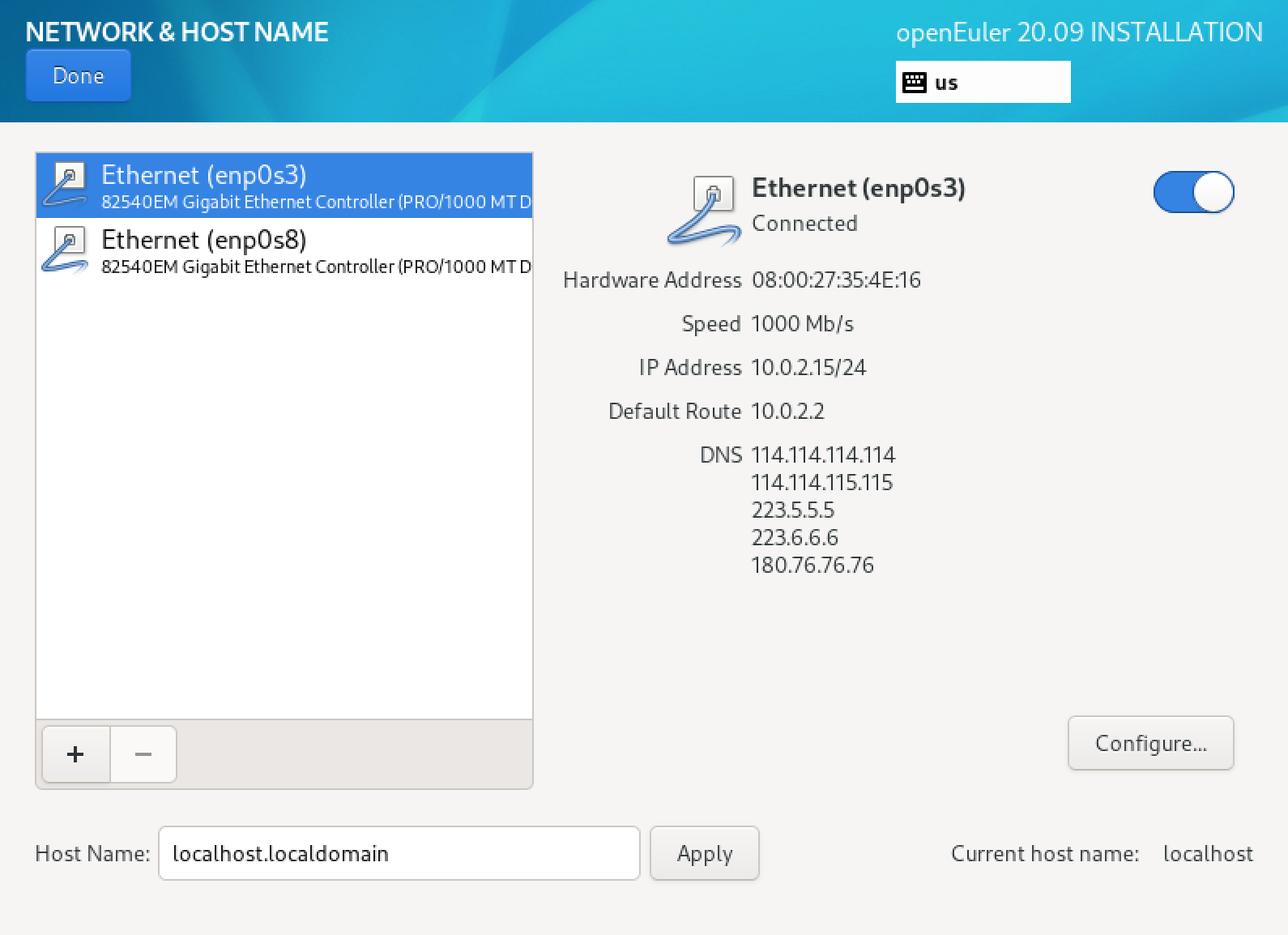
保持安装磁盘的默认设置：



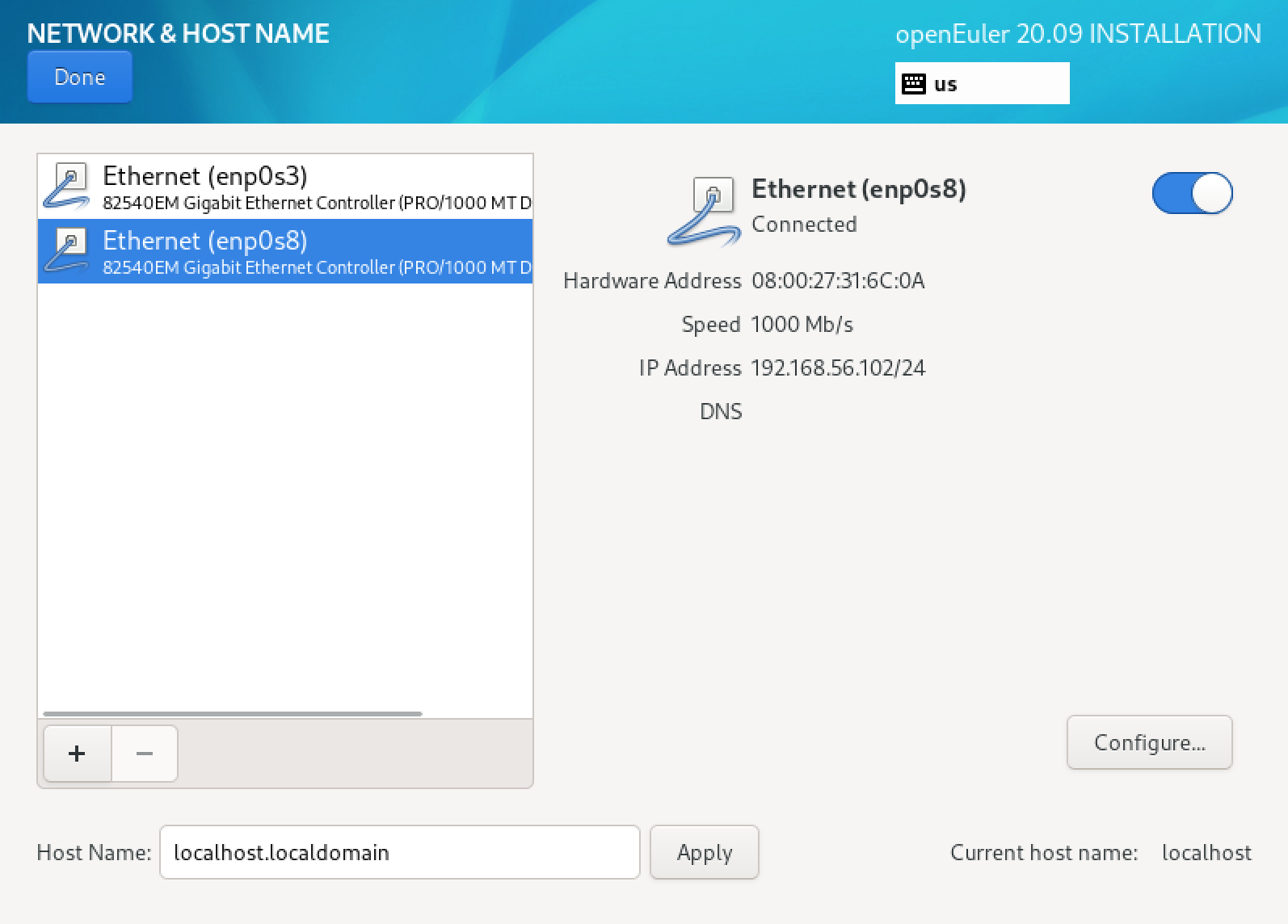
保持安装软件的默认设置：



正常情况下，第一张网卡connected：



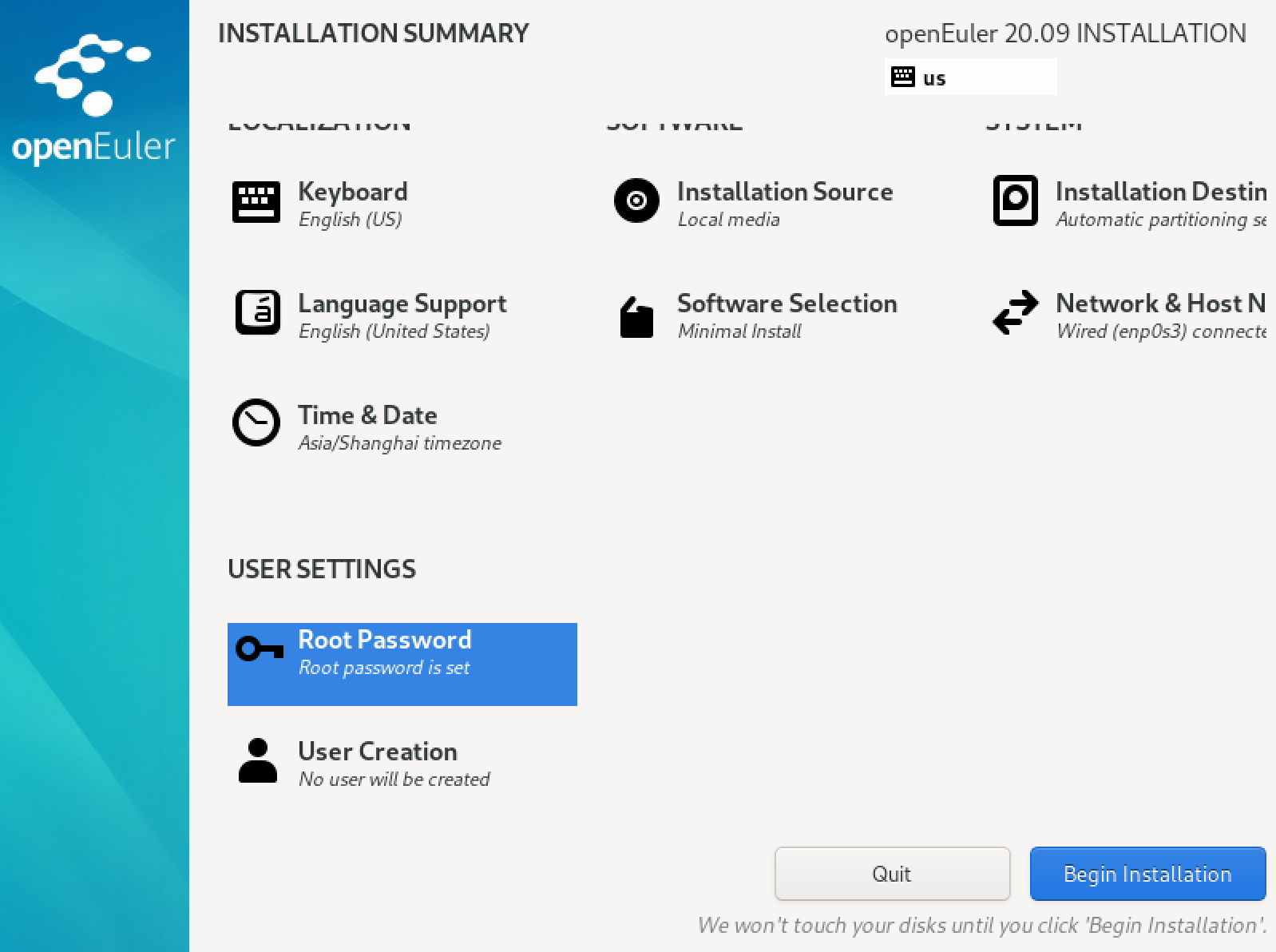
第二张网卡也connected：



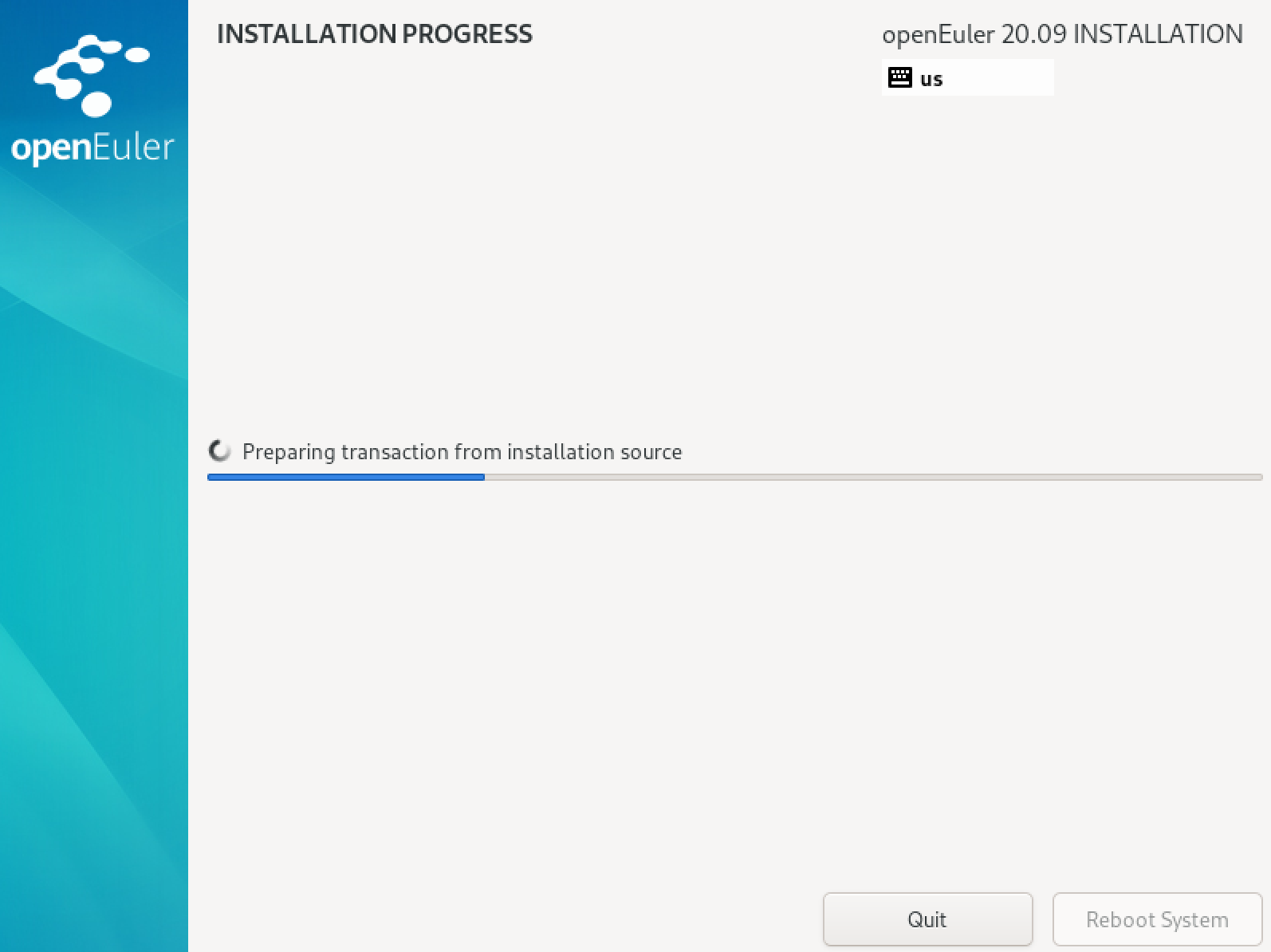
设置Root Password

注意：只需要设置root账户密码即可，无需再单独创建用户，root账户密码必须设置。

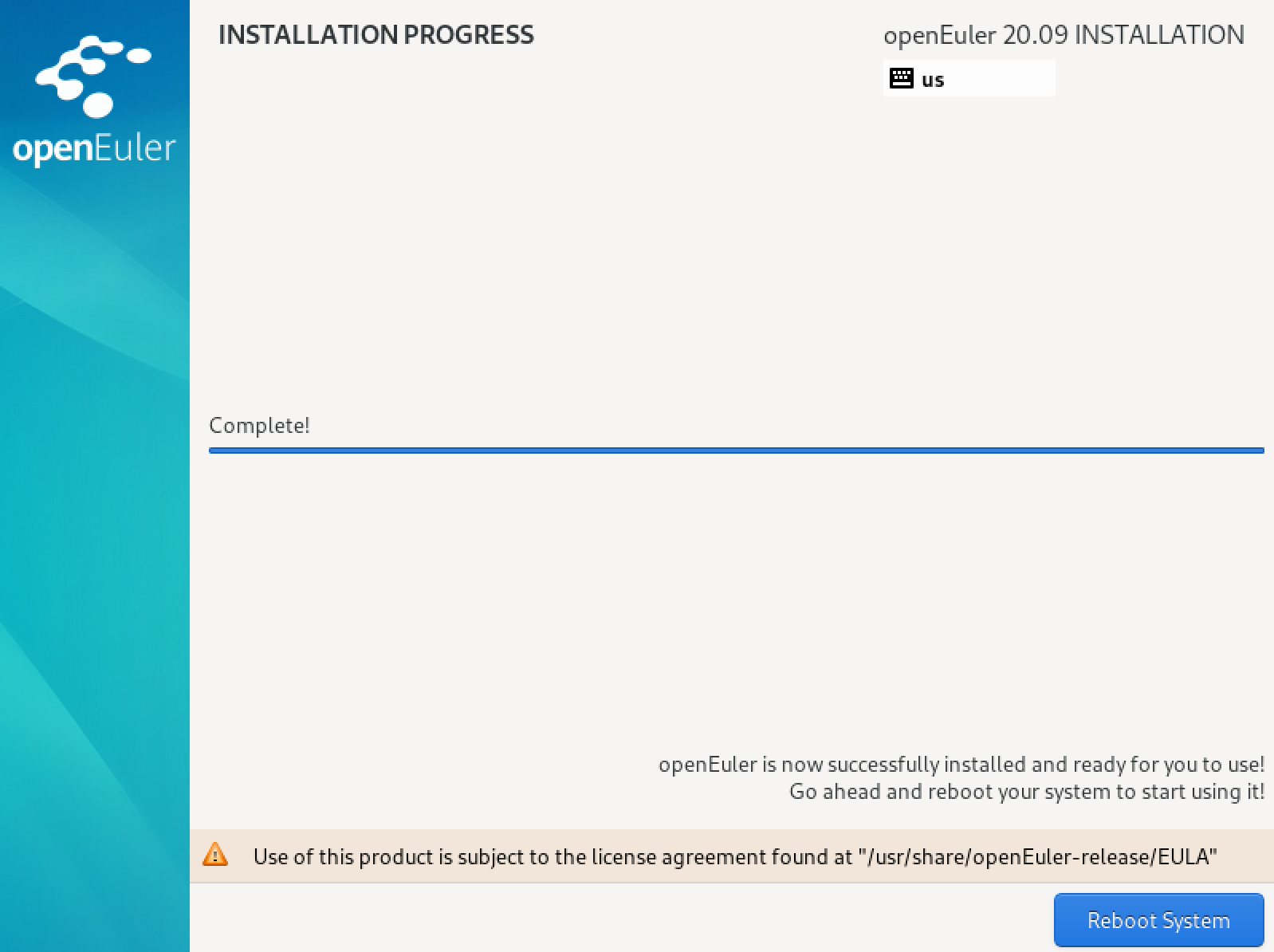
推荐密码：openeuler@123



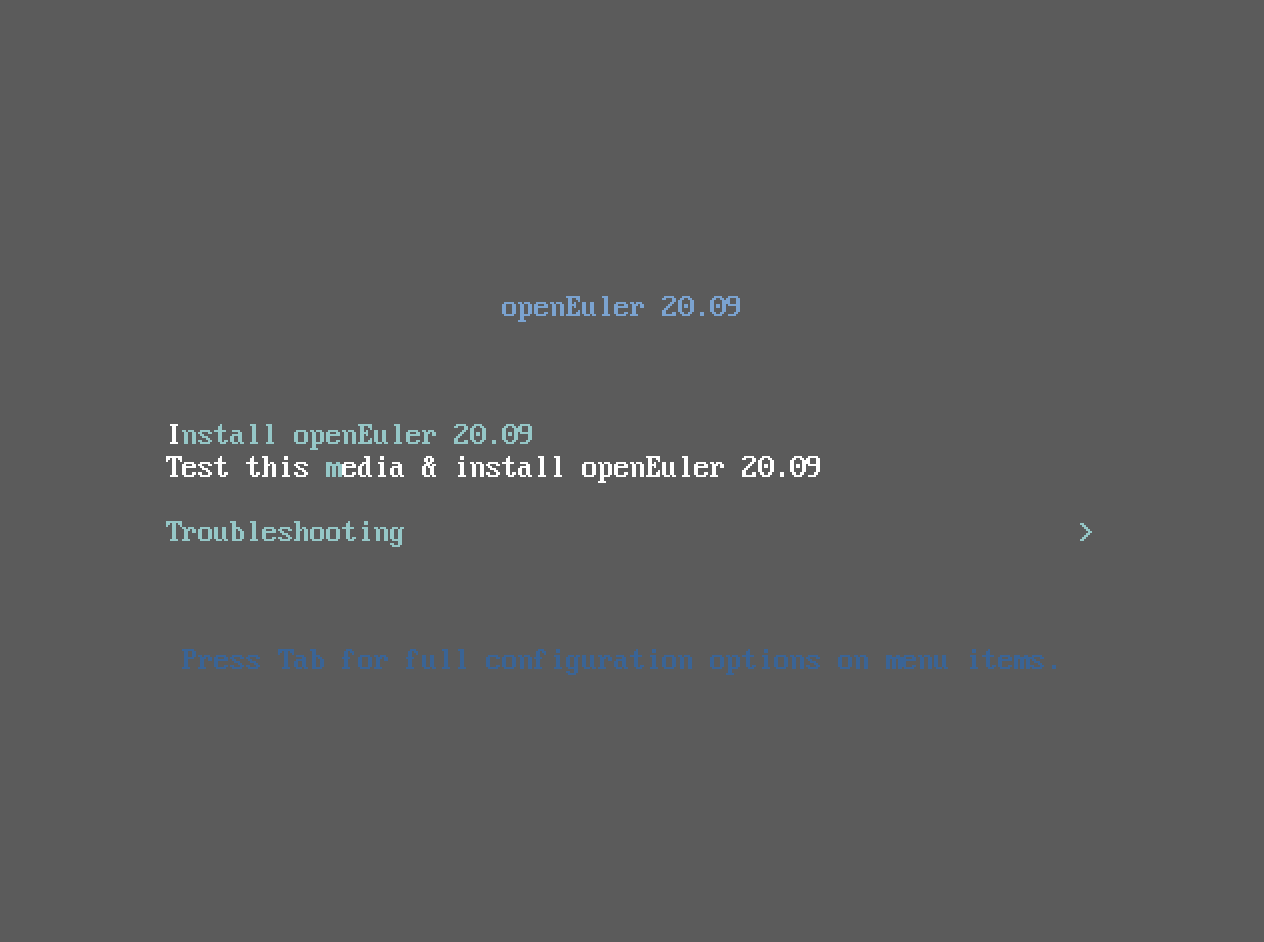
点击“开始安装”按钮开始安装：



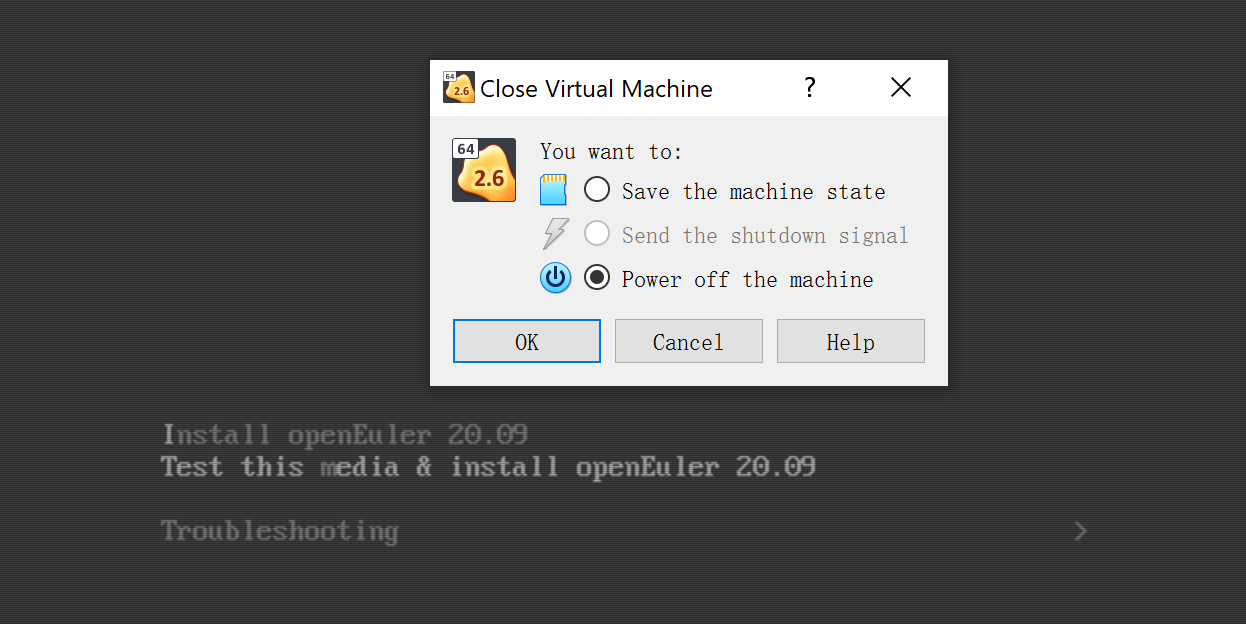
安装完成后点击“重启系统”按钮：



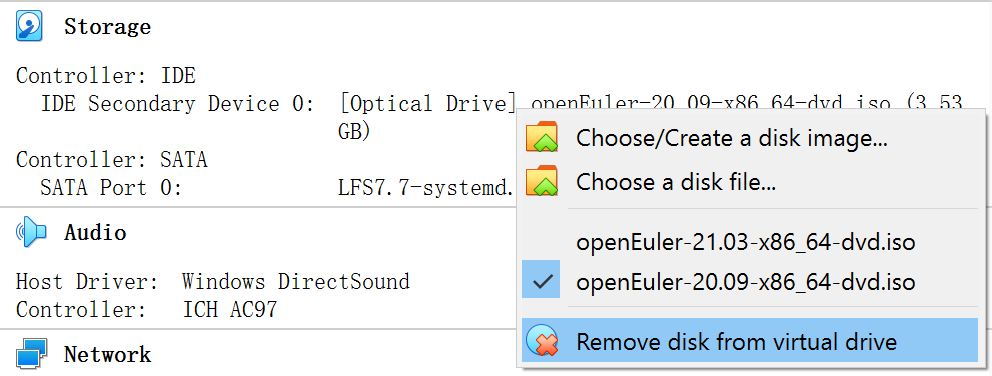
再次进入菜单选择界面时，移动键盘的上下键以将系统停留在该界面：



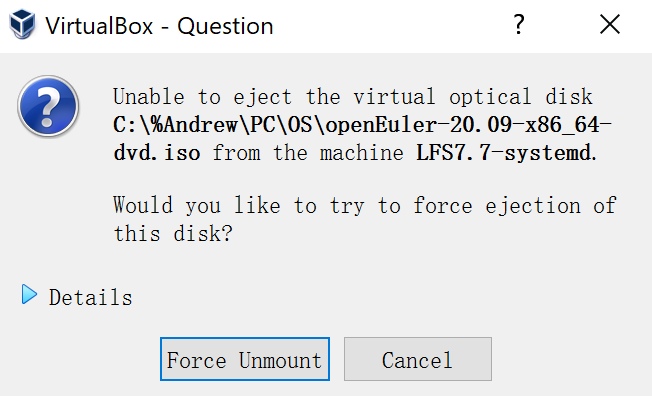
这时需要关机虚拟机：



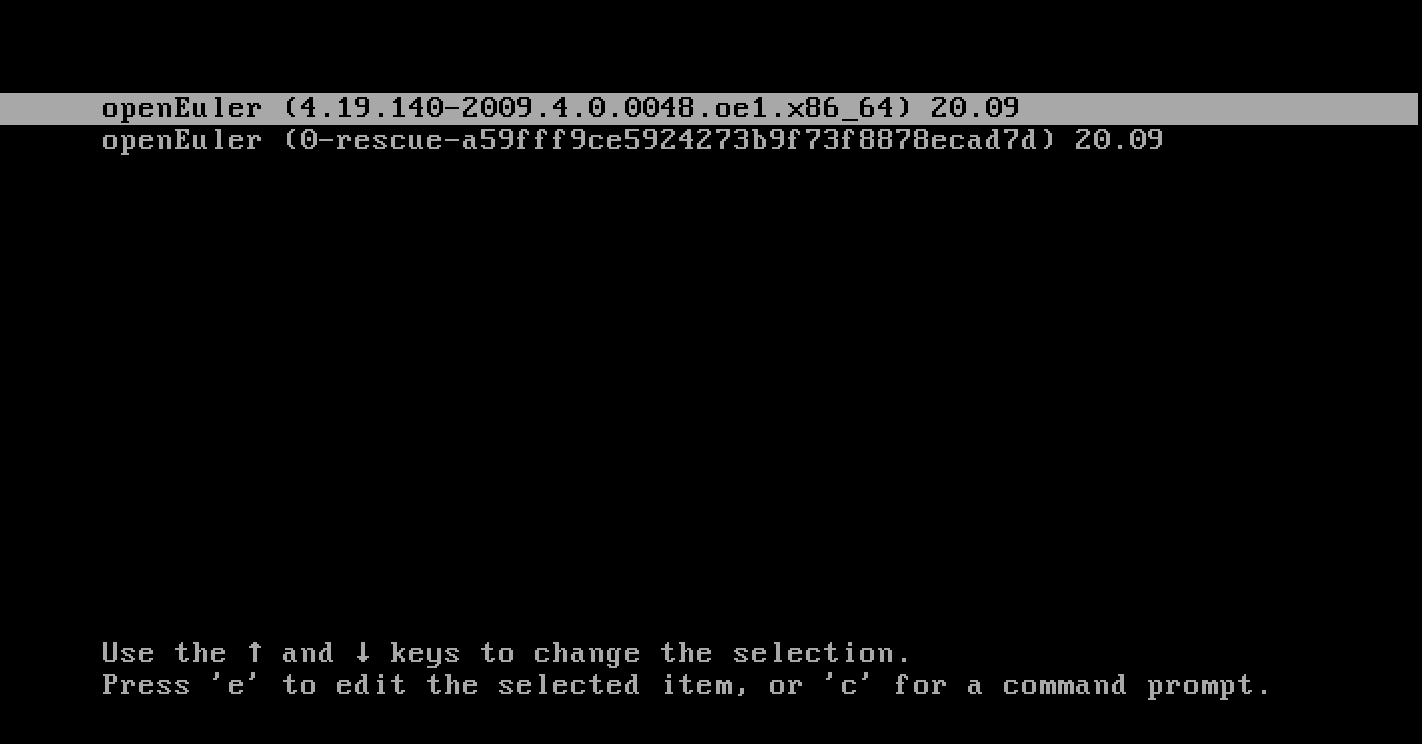
移出光驱中的镜像：



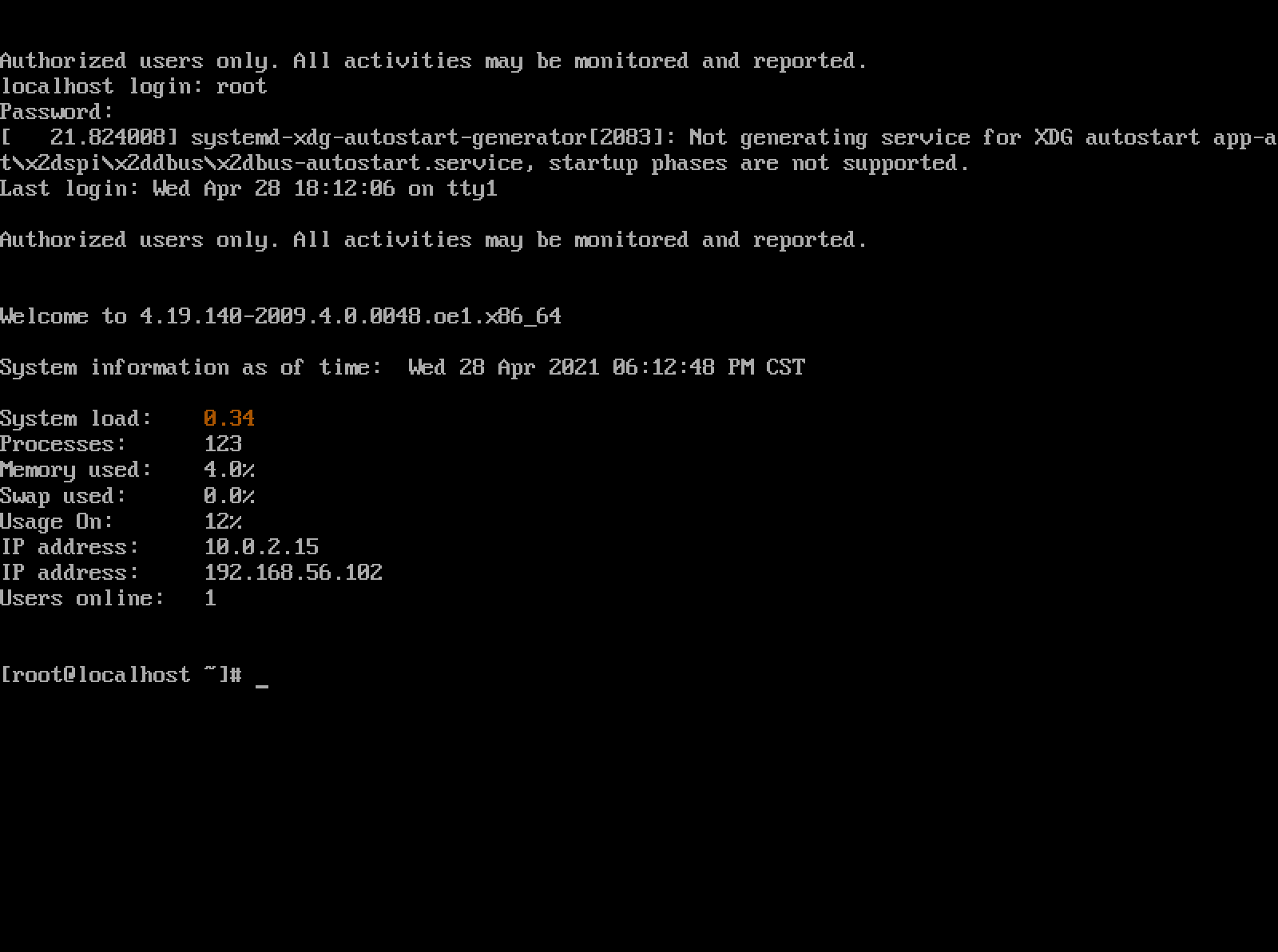
若提示是否强行移出则选择强行移出：



重启虚拟机，系统进入openEuler启动界面，用上下键选择第一个菜单项并按回车键启动（默认也是以第一个菜单项启动）：



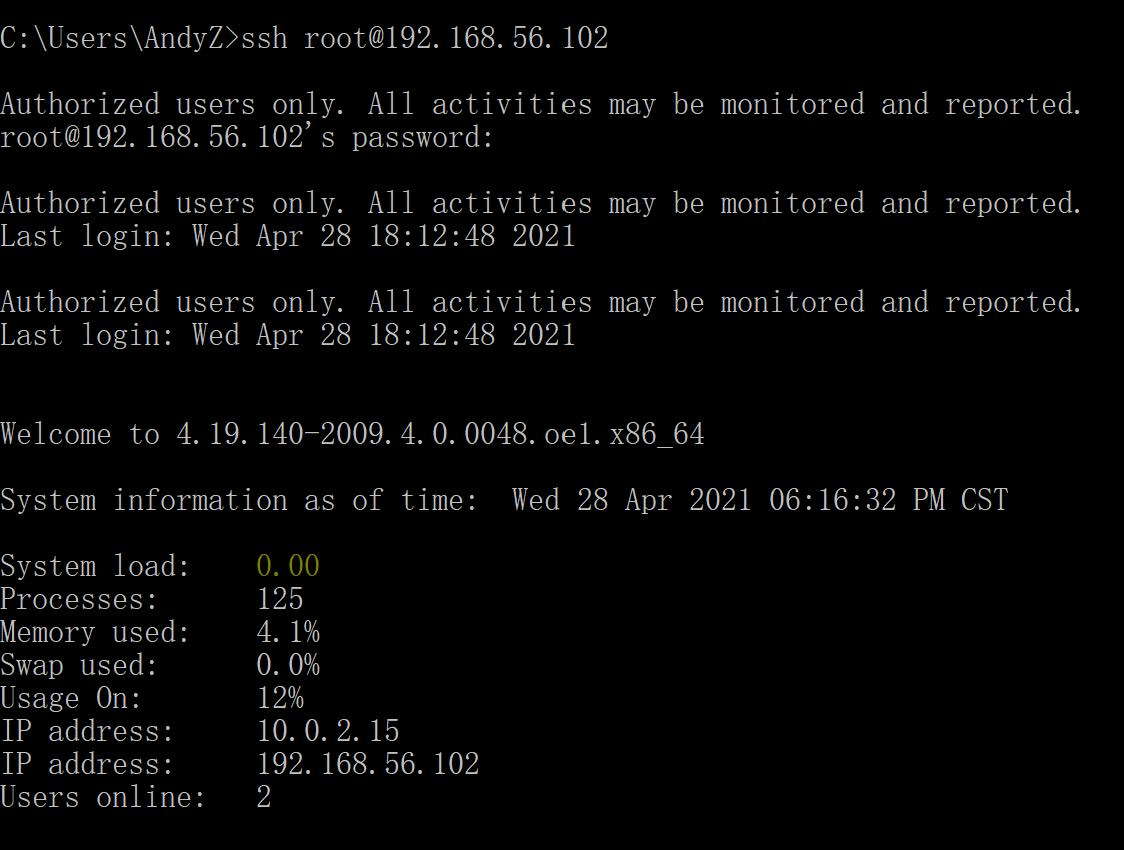
先在VirtualBox中以root用户登录openEuler操作系统：



为便于操作，再打开命令终端（如果是Windows 10操作系统，在键盘上按下“ +R”进入cmd命令行窗口；如果是macOS操作系统，在键盘上按下“ +空格键”，在弹出的搜索框中输入“Terminal.app”然后回车打开终端）通过ssh命令以root身份登录openEuler，如：

ssh root@*192.168.56.102*

注意：您应该用自己虚拟机的IP地址替换上述IP地址。



注意：以上登录过程中输入密码时不会有任何显示。

至此，已完成虚拟机安装，接下来的实验操作请参考shell脚本build-lfs7.7systemd.sh

## 准备实验环境

### 宿主机软件环境准备

在宿主机中安装必要的工具：

yum group install -y "Development Tools"

yum install -y bc

yum install -y openssl-devel

yum install -y texinfo # for makeinfo

yum install -y vim

请按“构建脚本”中创建的version-check.sh检测构建环境。结果可能如下：

bash, version 5.0.17(1)-release

/bin/sh -> /usr/bin/bash

Binutils: (GNU Binutils) 2.34

bison (GNU Bison) 3.6.4

yacc is /usr/bin/yacc - 1.9 20200330

bzip2, Version 1.0.8, 13-Jul-2019.

Coreutils: 8.32

diff (GNU diffutils) 3.7

find (GNU findutils) 4.7.0

GNU Awk 5.1.0, API: 3.0 (GNU MPFR 4.1.0, GNU MP 6.2.0)

/usr/bin/awk -> /usr/bin/gawk

gcc (GCC) 9.3.1

g++ (GCC) 9.3.1

(GNU libc) 2.31

grep (GNU grep) 3.4

gzip 1.10

Linux version 4.19.140-2009.4.0.0048.oe1.x86\_64 (abuild@ecs-obsworker-207) (gcc version 9.3.1 (GCC)) #1 SMP Thu Sep 24 09:39:46 UTC 2020

m4 (GNU M4) 1.4.18

GNU Make 4.3

GNU patch 2.7.6

Perl version='5.32.0';

sed (GNU sed) 4.8

tar (GNU tar) 1.32

texi2any (GNU texinfo) 6.7

xz (XZ Utils) 5.2.5

g++ compilation OK

这表明该系统满足构建LFS系统的需要。

另外，运行library-check.sh脚本可能会有如下结果：

libgmp.la: not found

libmpfr.la: not found

libmpc.la: not found

我们在此处忽略它，因为后面的步骤中会通过这些库的源码解决问题。

### 宿主机硬件环境准备

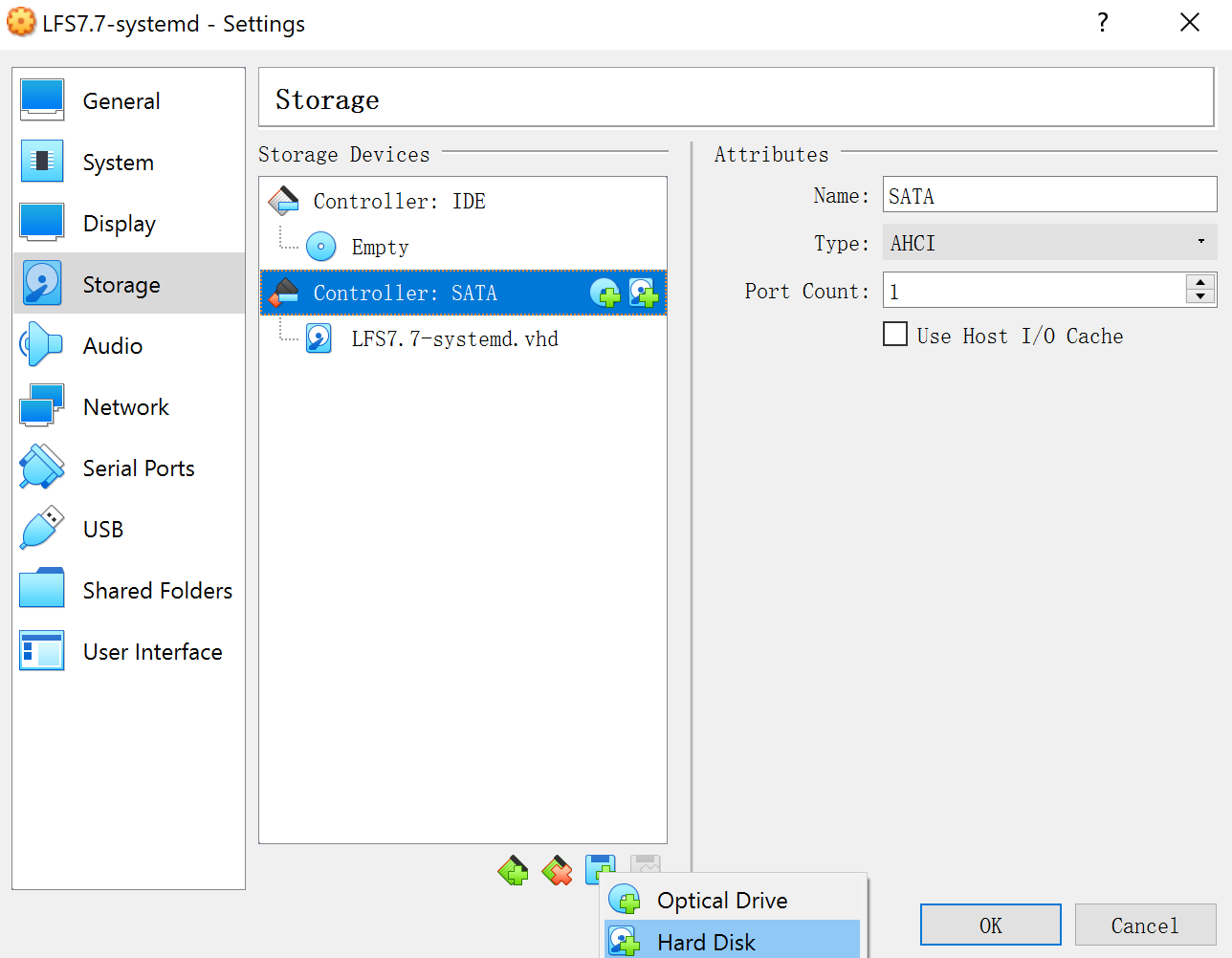
现在，我们准备一块新硬盘sdb以便在其上构建LFS系统，这样可以很方便地和（在sda上地）宿主系统分开。

首先，关闭虚拟机：

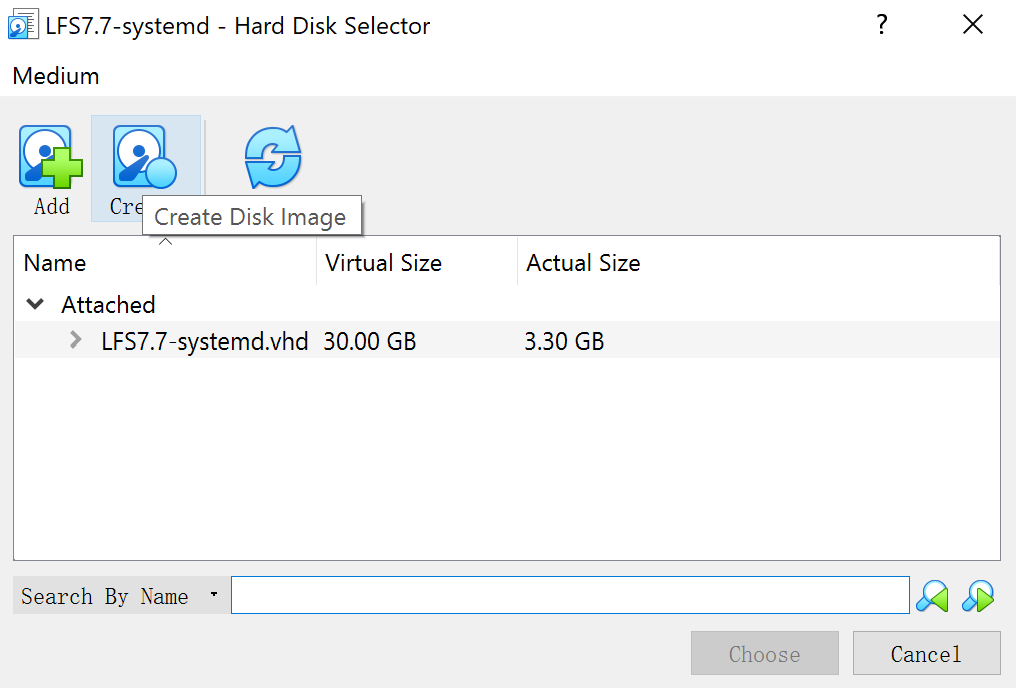
shutdown -h now

然后给虚拟机添加一块磁盘，其编号将自动为sdb。具体步骤如下：

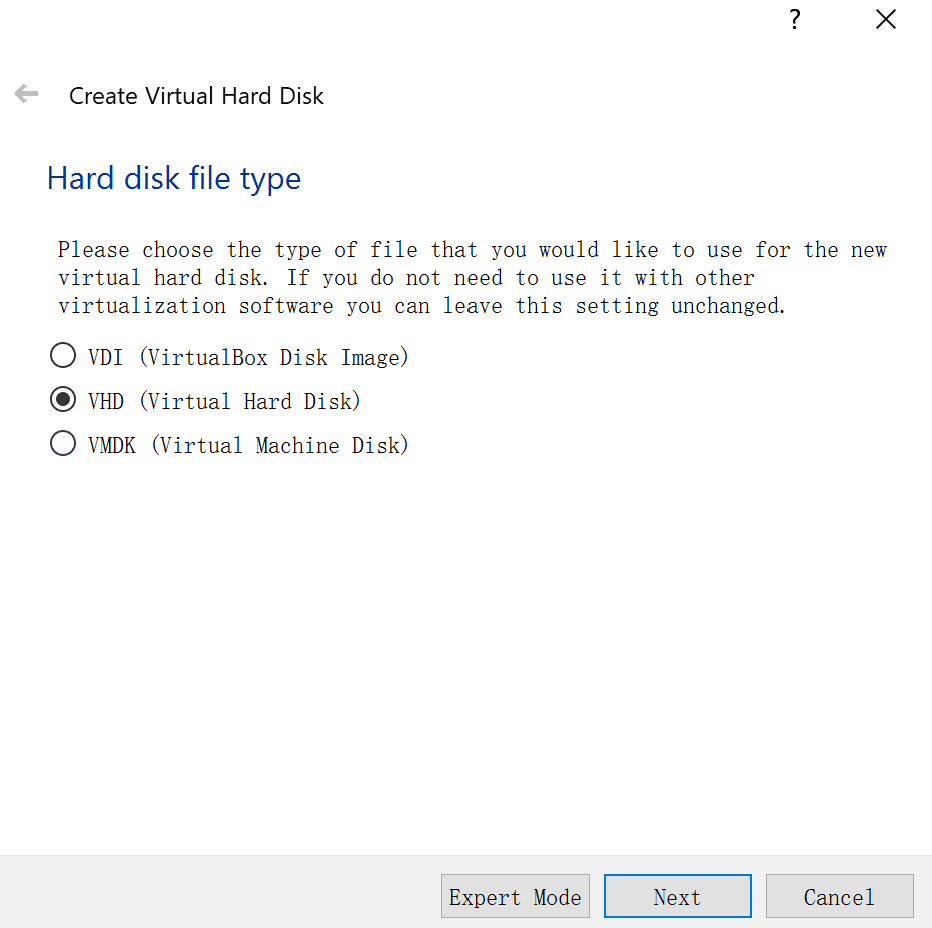
在VirtualBox中选择添加硬盘地菜单项：



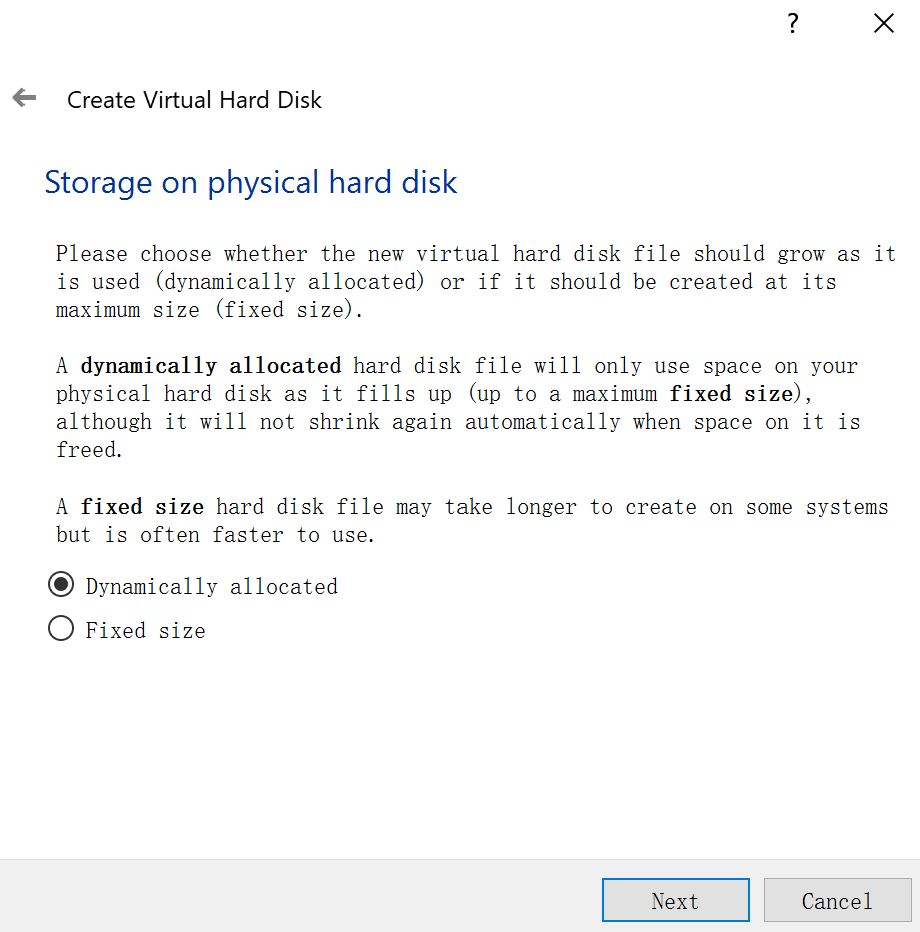
然后选择创建一张磁盘：



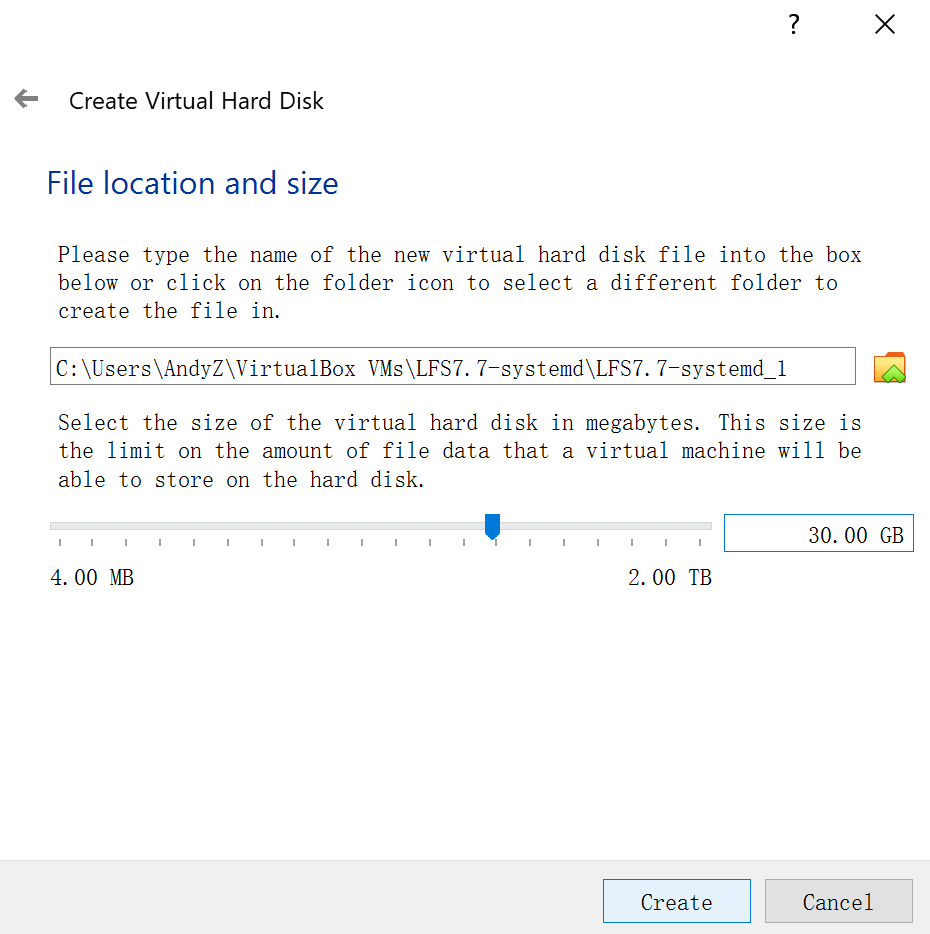
类型为VHD：



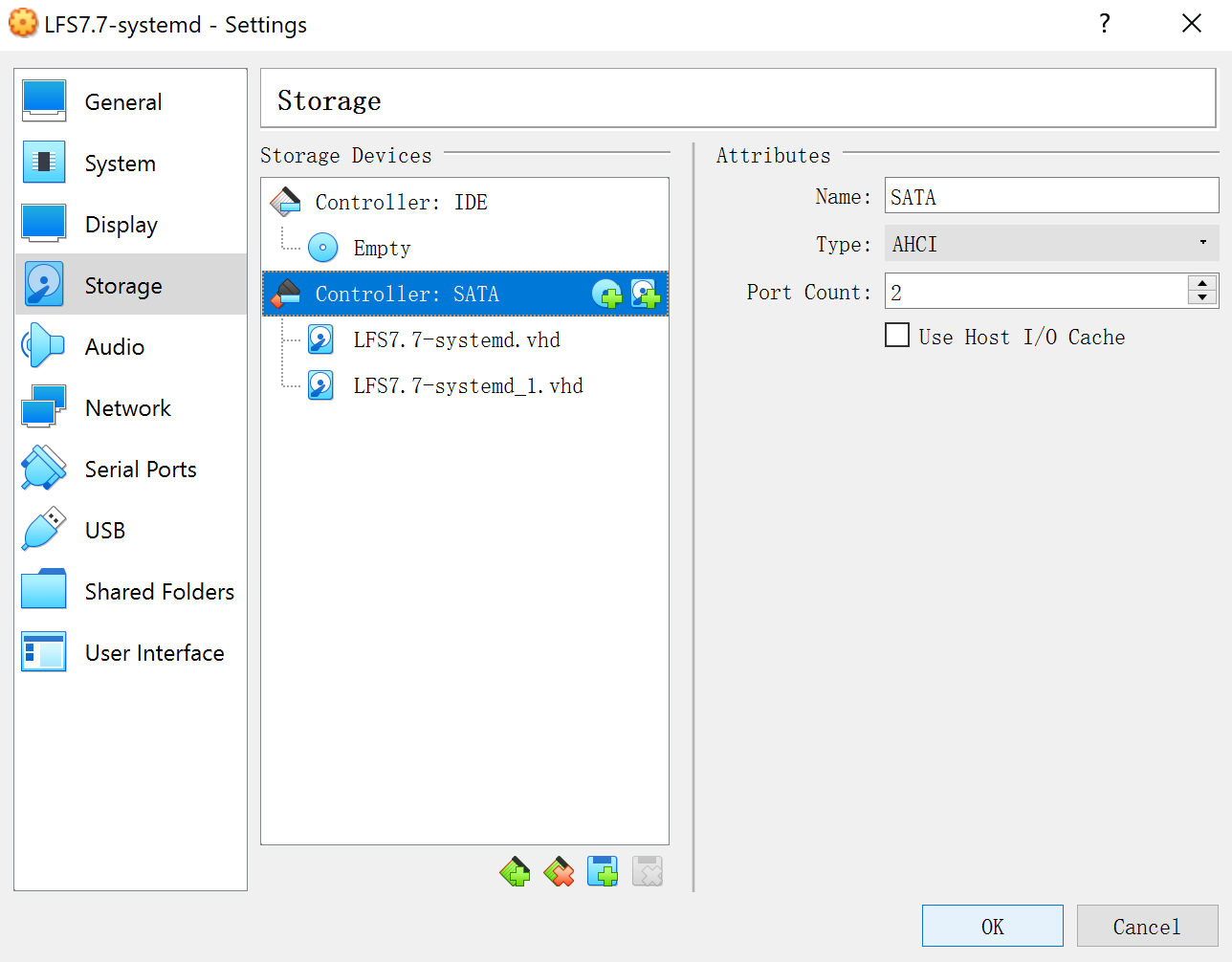
动态增长：



推荐大小30GB：



这是创建成功之后的显示结果：



添加好第二块硬盘之后我们可以通过以下命令查看磁盘信息：

lsblk

fdisk -l /dev/sdb

现在开始对该磁盘进行分区：

fdisk /dev/sdb

Welcome to fdisk (util-linux 2.35.2).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Device does not contain a recognized partition table.

Created a new DOS disklabel with disk identifier 0x9d1c2177.

Command (m for help): **n**

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p):

Using default response p.

Partition number (1-4, default 1):

First sector (2048-62914559, default 2048):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-62914559, default 62914559):

Created a new partition 1 of type 'Linux' and of size 30 GiB.

Command (m for help): **w**

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

请注意此处我们将整个硬盘sdb建成一个主分区sdb1。

然后对该磁盘进行格式化（该盘即是LFS目标系统将要被编译和安装的地方）：

mkfs -v -t ext4 /dev/sdb1

最后用以下命令检测结果：

lsblk

blkid

至此我们已经全部完成宿主机系统的准备工作。

# 从社区获取资料

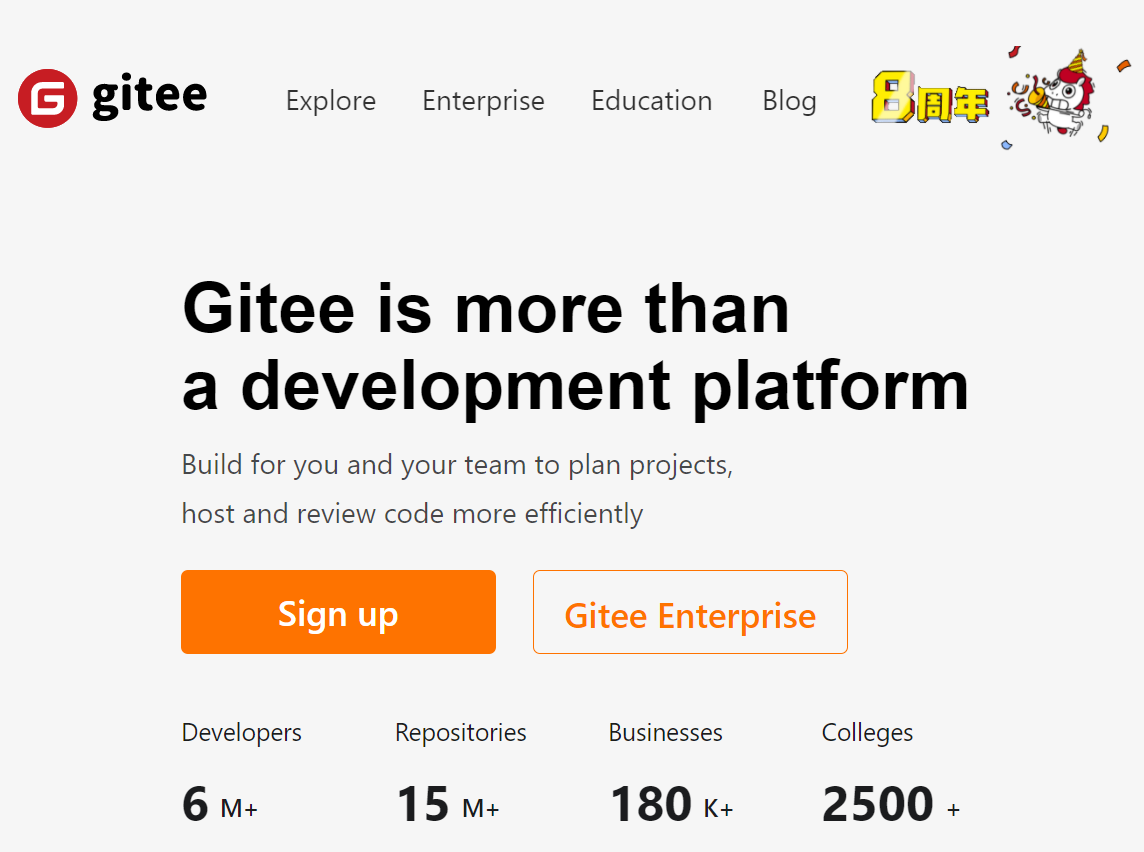
本实验的参考资料（包括本实验指导手册）是放在openEuler社区的代码托管地gitee.com上的。

## 注册gitee账户

打开gitee网站

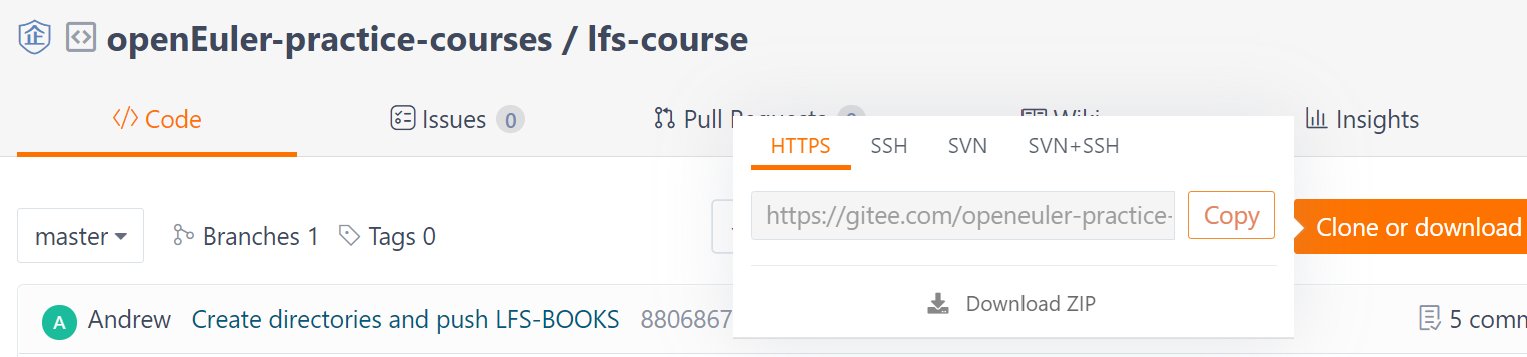
网站链接：<https://gitee.com/>

注册自己的gitee账户



点击“Sign up”进行注册。注意：最好以自己私人邮箱进行注册。注册完毕后以该账号登录gitee。

获取lfs-course的仓库地址



打开网址：<https://gitee.com/openeuler-practice-courses/lfs-course> 并复制该仓库的地址，其地址为：<https://gitee.com/openeuler-practice-courses/lfs-course.git>

## 从git仓库获取资料

Git的初始设置

若是第一次使用git命令，则要设置姓名和邮箱（请用英文输入）。

git config --global user.name "your-user-name"

git config --global user.email "your-email-address-on-gitee"

在刚刚安装的openEuler环境下运行以上命令。注意邮箱地址应与gitee账号的邮箱地址保持一致。

克隆lfs-course仓库

mkdir ~/openEuler

cd ~/openEuler

git clone https://gitee.com/openeuler-practice-courses/lfs-course

cd cd lfs-course/

ls

LICENSE README.en.md README.md lfs-7.7-systemd/

本次实验的资料即在“lfs-7.7-systemd/”目录里面。

## 将资料复制到本地

下面以在Windows 10中为例讲解如何将虚拟机中的资料拷贝到本地。

在虚拟机中查看IP地址

ip a | grep 192.168

inet *192.168.56.102*/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8

可以看出该虚拟机本地IP地址为*192.168.56.102*。

在Windows的File Explorer中打开Downloads文件夹

Graphical user interface, application, Word

Description automatically generated

这样保证文件夹路径中不包含空格或中文名。

在File Explorer的地址栏中输入“cmd”并按回车

这样会打开一个命令窗，并且路径就定位在Downloads文件夹下。

用scp命令进行拷贝

scp -r root@192.168.56.102:~/opc/lfs-course/ ./

这样会将虚拟机lfs-course目录中所有内容（包括目录本身）拷贝至本地（对于本实验，您也可以仅仅拷贝lfs-7.7-systemd/目录下内容）。

# LFS系统的构建

现在，我们就要开始真正构建LFS系统之旅啦！该过程分如下几个阶段：

1. 准备构建LFS系统所需的包和补丁
2. 创建并以lfs用户编译临时工具链
3. 在chrooted环境下构建真正的LFS目标系统
4. 配置LFS目标系统
5. 编译并安装内核
6. 设置GRUB使新系统可引导
7. 进入新系统

这里只列出一些特别注意的地方，具体构建过程请参考“构建脚本”和LFS-BOOK-7.7-systemd.pdf。

## 准备构建LFS系统所需的包和补丁

请在LFS官网下载页面（https://www.linuxfromscratch.org/lfs/download.html）找到“Packages for LFS”一节并将lfs-packages-7.7-systemd.tar包下载到您的本地。

## 创建并以lfs用户编译临时工具链

### /mnt/lfs及lfs用户

为了避免损坏宿主系统，我们将建立lfs用户并以他编译临时工具链。

我们首先设置了$LFS环境变量：

export LFS=/mnt/lfs

然后在宿主机上创建相应文件夹（这个文件夹将成为chrooted的根）：

mkdir -pv $LFS

并将LFS系统将要编译和安装到的那个盘挂载进来（通过修改宿主机的fstab）：

/dev/sdb1 /mnt/lfs ext4 defaults 1 1

创建lfs用户：

groupadd lfs

useradd -s /bin/bash -g lfs -m -k /dev/null lfs

### $LFS/sources 和$LFS/tools文件夹

我们用scp命令将构建LFS系统需要的包从本地拷贝到/mnt/lfs：

scp lfs-packages-7.7-systemd.tar root@*192.168.11.130*:/mnt/lfs/

（注意：请替换命令中的IP地址为您自己宿主机的IP地址。）

这个包将会以lfs用户解压到$LFS/sources。

另外请注意临时工具链都会安装到$LFS/tools。

### 通用编译指南

对于每个软件包：

a. 以lfs用户解压要编译的软件包

b. 进入到解压后创建的目录中

c. 根据指南说明编译软件包

d. 回退到源文件目录

e. 除非特别说明，删除解压出来的目录（和为了编译而创建的build目录）

### gcc-4.9.2编译问题

对于gcc-4.9.2编译过程中的一个问题（和cfns.h头文件有关），这里非官方解决方案安装一个补丁：

cd $LFS/sources/gcc-4.9.2

patch -p1 < cfns-4.9.2.patch

提示：补丁“cfns-4.9.2.patch”可以克隆创新实践课的仓库得到：

git clone https://gitee.com/openeuler-practice-courses/lfs-course.git

在GCC后续版本中，通过增加一个新的类解决了此问题。当然，您也可以提供您自己的解决方案。

### 其他需要注意的地方

Stripping节约的空间有限，可以不做。

注意几次chmod和chown的作用。

## 在chrooted环境下构建真正的LFS目标系统

和LFS-BOOK不同的是，我们允许您不断地重启宿主机系统，这在“构建脚本”中多次重启的命令也可以看出来。但在此节每次重启系统后，需要重新mount和chroot：

### mount & chroot

本节中的所有安装工作都是在以root用户登录宿主机系统后挂载相应目录、再进入chroot环境进行的。为了可以在宿主机reboot后接着进行LFS系统的构建工作，建议将每次root登录进来之后的mount和chroot写成脚本：

mount脚本：

# For mount-and-populate later - step 1/2

cat > ~/mount-and-populate.sh << "EOF"

#!/bin/bash

# 6.2.2. Mounting and Populating /dev

mount -v --bind /dev $LFS/dev

# 6.2.3. Mounting Virtual Kernel File Systems

mount -v --bind /dev/pts $LFS/dev/pts -o gid=5,mode=620

mount -vt proc proc $LFS/proc

mount -vt sysfs sysfs $LFS/sys

mount -vt tmpfs tmpfs $LFS/run

EOF

sh ~/mount-and-populate.sh

chroot脚本：

cat > ~/chroot-lfs.sh << "EOF"

#!/bin/bash

# 6.4. Entering the Chroot Environment

chroot "$LFS" /tools/bin/env -i \

HOME=/root \

TERM="$TERM" \

PS1='\u:\w\$ ' \

PATH=/bin:/usr/bin:/sbin:/usr/sbin:/tools/bin \

/tools/bin/bash --login +h

EOF

# chroot - step 2/2

sh ~/chroot-lfs.sh

### 通用编译指南

对于每个软件包：

a. 进入chroot环境后解压要编译的软件包

b. 进入到解压后创建的目录中

c. 根据指南说明编译软件包

d. 回退到源文件目录

e. 除非特别说明，删除解压出来的目录（和为了编译而创建的build目录）

### 第二个版本的chroot

完成本节的编译安装工作后，由于不再需要/tools目录，在以后的任务中可以使用如下的（mount和）chroot命令：

# mount-and-populate - step 1/2

sh ~/mount-and-populate.sh

cat > ~/chroot-lfs2.sh << "EOF"

#!/bin/bash

# 6.72. Cleaning Up

chroot "$LFS" /usr/bin/env -i \

HOME=/root \

TERM="$TERM" \

PS1='\u:\w\$ ' \

PATH=/bin:/usr/bin:/sbin:/usr/sbin \

/bin/bash --login

EOF

# chroot2 - step 2/2

sh ~/chroot-lfs2.sh

注意这里mount命令脚本还是和以前一样的，但是chroot脚本的内容与前面的有所不同。

### 其他问题

LFS-BOOK 6.25.3. Setting the root password那一节以passwd root配置的密码即LFS系统的root密码。

由于节约不了多少空间，cleaning up问题被再次忽略。

## 配置LFS目标系统

本节为LFS系统进行网络、主机名、/etc/hosts文件的设置，其余的配置可以忽略。注意本节并没有配置一个真正可以访问互联网的网络，这一任务留在进阶实验中由学生们自己去探索。

## 编译并安装内核

本节进行内核的编译和安装。需要注意的是make menuconfig这一步，除LFS-BOOK中所提到的那些配置选项外，需要选择和虚拟机提供的硬件相匹配的驱动（特别是硬盘的驱动），否则以后启动新系统时系统将会mount不上sdb。

比如说，如果您安装了SCSI的硬盘，则需要选择以下配置：

# For SCSI Disk

Linux Kernel Configuration

-> Device Drivers

-> SCSI device support

-> SCSI disk support

同时也建议选中以下配置：

# For BusLogic

Linux Kernel Configuration

-> Device Drivers

-> SCSI device support

-> SCSI low-level drivers

-> BusLogic SCSI support

# For LSI Logic

Linux Kernel Configuration

-> Device Drivers

-> Fusion MPT device support

-> Fusion MPT (base + ScsiHost) drivers

（若您是按本文指导配置的虚拟机，则以上三个项目的配置保持其默认值即可。）

这里列出需按LFS-BOOK中建议进行的配置：

# Refer to LFS-BOOK

General setup --->

[\*] open by fhandle syscalls [CONFIG\_FHANDLE]

[ ] Auditing support [CONFIG\_AUDIT]

[\*] Control Group support [CONFIG\_CGROUPS]

Processor type and features --->

[\*] Enable seccomp to safely compute untrusted bytecode [CONFIG\_SECCOMP]

Networking support --->

Networking options --->

<\*> The IPv6 protocol [CONFIG\_IPV6]

Device Drivers --->

Generic Driver Options --->

[ ] Support for uevent helper [CONFIG\_UEVENT\_HELPER]

[\*] Maintain a devtmpfs filesystem to mount at /dev [CONFIG\_DEVTMPFS]

[ ] Fallback user-helper invocation for firmware loading [CONFIG\_FW\_LOADER\_USER\_HELPER]

Firmware Drivers --->

[\*] Export DMI identification via sysfs to userspace [CONFIG\_DMIID]

File systems --->

[\*] Inotify support for userspace [CONFIG\_INOTIFY\_USER]

<\*> Kernel automounter version 4 support (also supports v3) [CONFIG\_AUTOFS4\_FS]

Pseudo filesystems --->

[\*] Tmpfs POSIX Access Control Lists [CONFIG\_TMPFS\_POSIX\_ACL]

[\*] Tmpfs extended attributes [CONFIG\_TMPFS\_XATTR]

## 设置GRUB使新系统可引导

### LFS系统GRUB设置

LFS系统（即目前的chrooted环境）可以按以下命令生成grub.cfg文件：

grub-install /dev/sdb

grub-mkconfig -o /boot/grub/grub.cfg

### 宿主机GRUB设置

对于宿主机的GRUB设置，将LFS系统的grub.cfg中menuentry模块复制到宿主机/boot/grub2/grub.cfg中的相应位置即可。

比如说，$LFS/boot/grub/grub.cfg的第一menuentry快的包含如下内容：

### BEGIN /etc/grub.d/10\_linux ###

menuentry 'GNU/Linux' --class gnu-linux --class gnu --class os $menuentry\_id\_option 'gnulinux-simple-*cc2f6dd5-caf9-4e91-9eac-edbfa484a4bc*' {

load\_video

insmod gzio

insmod part\_msdos

insmod ext2

set root='hd1,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd1,msdos1 --hint-efi=hd1,msdos1 --hint-baremetal=ahci1,msdos1 cc2f6dd5-caf9-4e91-9eac-edbfa484a4bc

else

search --no-floppy --fs-uuid --set=root *cc2f6dd5-caf9-4e91-9eac-edbfa484a4bc*

fi

echo 'Loading Linux 3.19-lfs-7.7-systemd ...'

linux /boot/vmlinuz-3.19-lfs-7.7-systemd root=/dev/sdb1 ro

}

注意其中的ID号和您的不会是一样的。然后把这部分内容增加到宿主机文件/boot/grub2/grub.cfg的对应位置，如这条语句的后面：

### BEGIN /etc/grub.d/10\_linux ###

当然，您可更改这条菜单项的标题GNU/Linux，使之更个性化一点。

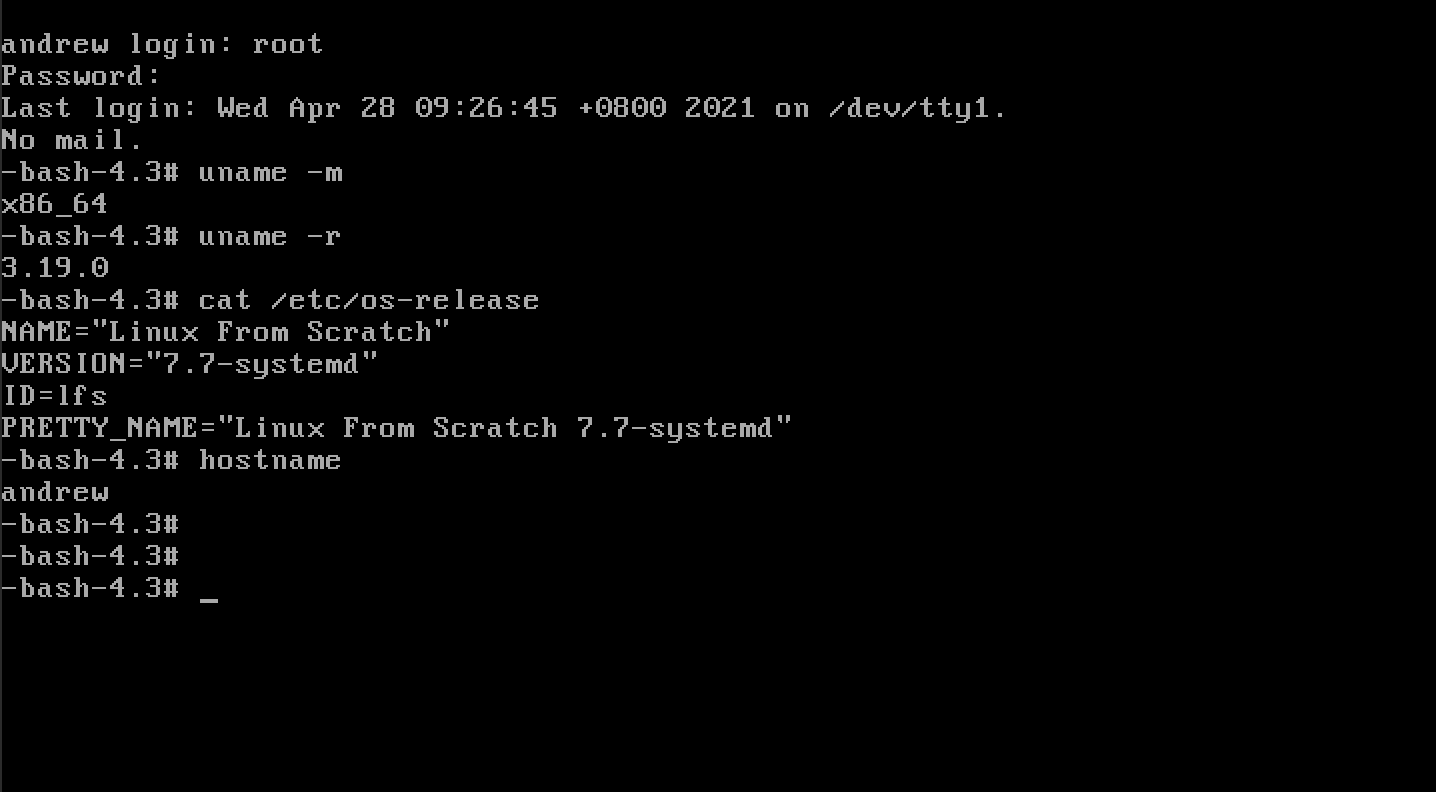
## 进入新的系统

重启系统，会出现LFS系统的选择项，移动键盘的上下键选中它，按回车键进入，

Text

Description automatically generated

以root用户登录，输入LFS-BOOK 6.25.3. Setting the root password节设置的密码。

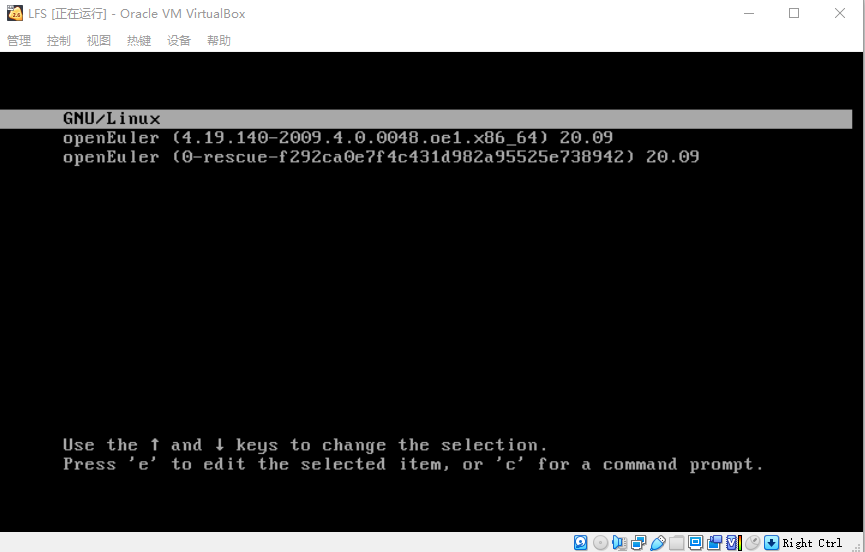


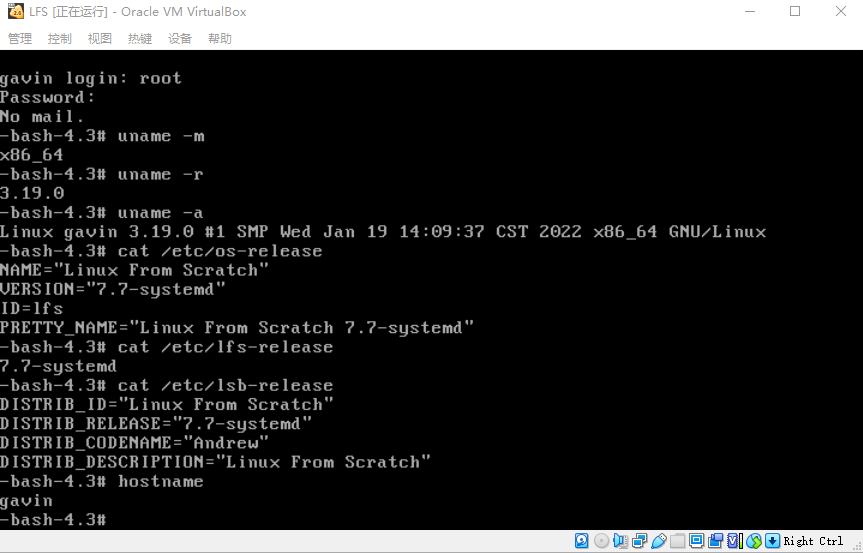
以上是进入新系统的情形。

至此大功告成！

# 作业的提交

本课程要求学生撰写课程设计报告，对构建成功的系统进行截图（一个是系统的启动页面，一个是系统登录后的执行页面，如下所示），对构建过程进行分析，并对课程设计进行心得体会。





# 缩略语表

|  |  |  |
| --- | --- | --- |
| **缩略语** | **英文全称** | **中文释义** |
| ABI | Application Binary Interface | 应用程序二进制接口 |
| LFS | Linux From Scratch | 白手起家构建Linux |
| LSB | Linux Standard Base | Linux标准规范 |
| POSIX | Portable Operating System Interface for UNIX | UNIX可移植操作系统接口 |
| VFS | Virtual File System | 虚拟文件系统 |